CMake » Mono » PAM » Расширения WordPress

LXF DVD

Xubuntu

плюс: **Smoothwall**

OpenOffice.org 2.3

» Coding для разработчика!

» X.org 7.3 и многое другое!

LINUX FORMAT

Главное в мире Linux



Декабрь 2007 № 12 (99)

LXF99 LXF Декабрь 2007 № 12 (99)

10 проектов «Сделай сам» » Xubuntu » Сравнение календарей » CMake » Патчи ядра » Прими участие » Секреты поиска

10 проектов «СДЕЛАЙ САМ»

» Смените прошивку в маршрутизаторе » Сделайте загружаемый USB-брелок » Соберите ферму для рендеринга » Создайте систему безопасности с web-камерами » И так далее!

Примите участие

Простые способы внести вклад в сообщество Open Source с. 32

38 страниц учебников на любой вкус »

Не пропусти свидание!
Лучшие программы-календари – лицом к лицу с. 16

Инъекция для ядра
Наложите все патчи, чтобы быть на острие атаки с. 74

WordPress
Десятка лучших расширений с. 51

Каталог агентства «РОСПЕЧАТЬ» – подписной индекс 20882
Каталог «ПРЕССА РОССИИ» – подписной индекс 87974

«Мы всегда хотели позиционировать себя как платформу для инноваций»

Джек Эбаутбул посланник Fedora с. 24





На линуксовке...

В этом выпуске мы особенно усердно популяризируем Open Source. Поэтому вопрос к команде такой: что, по-вашему, придало бы перцу обычной встрече местного LUG'a?



Пол Хадсон
Групповое караоке. Команды выбирают по файлу со своих ПК и поют его на мотив Бритни Спирс. Спевшему **/dev/urandom** – бонус.



Грэм Моррисон
Обвините двух членов LUG в связи с моей женой. Или с женами друг друга. Или друг с другом.



Майк Сондерс
Зудящий порошок, вопящие подушки, дымовые шашки и ругательства по телефону от короля архипелага Тонга.



Эфраин Эрнандес-Мендоса
Фахита и стриптизерша? Погодите... Энчилада и стриптизер, нет? А, понял, сигары «Коронас» и стриптиз? НЕТ?! Ну, я не знаю...



Мэтт Нейлон
Пригласите Стефена Фрая и покажите ему, почему он не так уж и умен. Мы проучим этого яблочника! www.stephenfry.com/blog/?p=3



Энди Ченнел
Вызовите Джерри Холлиуэлл и Эмму Бантон, пусть явятся в бикини из листьев кориандра.



Энди Хадсон
Пригласите разом «Партию Единой Родины и Единых Целей», садомазошоу-герлз и представителей районного отдела народного образования!



Д-р Крис Браун
Живые дебаты между Стивом Баллмером и Ричардом Столлменом. Только убедитесь, что хватит свободных стульев.



Нейл Ботвик
Группа *Uriah Heep* вначале называлась *Spice*: отвязная сессия песен из их *Very 'Eavy... Very 'Umble* была бы просто великолепна.



Дэвид Картрайт
Попросите BBC Click прислать Кейт Рассел. Это будет полезно для тех участников, кто редко бывает на людях, и вообще – она тоже немного технарь.



Сделай сам – 11

В спецрепортаже этого месяца мы предлагаем вашему вниманию 10 способов скоротать долгий зимний вечер, зачастую – даже с пользой для себя. Кроме того, одной из центральных тем данного номера является соучастие (см. стр. 32), и я подумал: а почему, собственно, это должны обязательно быть два различных материала? Поэтому, пользуясь случаем, предлагаю вам одиннадцатый проект: «Сделаем мир Linux лучше»! Требуется: компьютер с Linux, желательно с подключением к Сети, и горячее желание изменить окружающую реальность. Время: ровно столько, сколько вы отведете.

Посмотрите внимательно на свой дистрибутив. Возможно, он уже идеален, поскольку, поставив его на компьютер, вы все время что-то шлифовали, докачивали, настраивали. Подумайте: может быть, кое-что из наработанного пригодится и другим людям? Если же на гладкой лакированной поверхности вашей системы остались сучки и задоринки – устраните их, а изменениями снова поделитесь с общественностью. Подпишитесь на списки рассылки, зарегистрируйте сообщение об ошибке и приложите свою заплатку в Bugzilla – и, при некотором везении, ваше имя вскоре будет красоваться в списке участников известного проекта. Да, на это может уйти больше одного вечера, но зато и «дивиденды» будут куда крупнее. LXF

Валентин Синицын » Главный редактор info@linuxformat.ru

Миссия журнала



- Пропаганда свободного ПО в России
- Продвижение решений с открытым кодом в бизнес-сообществе
- Поддержка российского Open Source сообщества
- Организация трибуны для разработчиков свободного ПО
- Обратная связь между разработчиками и потребителями ПО

Как с нами связаться

Письма для публикации: letters@linuxformat.ru
Подписка и предыдущие номера: subscribe@linuxformat.ru
Техническая поддержка: answers@linuxformat.ru
Проблемы с дисками: disks@linuxformat.ru
Общие вопросы: info@linuxformat.ru
Web-сайт: www.linuxformat.ru
› **Адрес редакции:** Россия, Санкт-Петербург, ул. Гончарная, 23, офис 54.
› **Телефон редакции:** (812) 717-00-37. Дополнительная информация на стр.128



Декабрь 2007 № 12 (99)

Содержание

Весь номер – прямо как на ладони: приятного чтения!

Учебники



LXF DVD99

Xubuntu 7.10
Любите Ubuntu, но предпочитаете что-то побыстрее? Мы уверены, эта разновидность всемирно популярного дистрибутива, основанная на Xfce, порадует ваш ПК.

SmoothWall 3.0
Не тратьте время на переделку дистрибутива общего назначения в межсетевой экран – SmoothWall Express изумительно справится с этой задачей прямо из коробки.

Мегапак средств разработки
Руки на стол! – с клавиатурой: мы приготовили для вас подборку языков программирования, включая Eiffel, Lisp, Haskell и даже COBOL!



› Если Xubuntu для вас слишком тяжел, ищите DSL на стр. 58.

Что за штука… CMake?

с. 30

Scribus уже использует это, а вам следует приступить прямо сегодня





LXFHotPicks



› Варвары из *WideLands* надерут вам…

LXF ИНТЕРВЬЮ

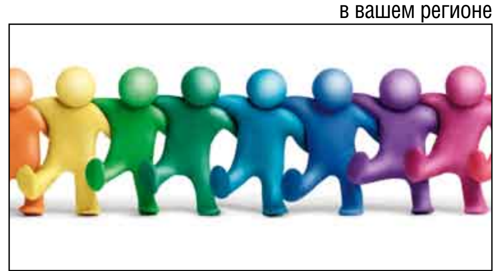
«На данный момент мы пребываем в некой сумеречной зоне.»

Джек Эбаутбул из Fedora с. 24

Прими участие

с. 32

Пять суперидей для пропаганды Linux в вашем регионе





Подпишись на Linux Format и сэкономь!



Содержание

LXFDVD внутри!

Хотите узнать, что на нем? См. с. 104

Спецрепортаж

10 проектов «СДЕЛАЙ САМ»

Создавайте 3D-изображения, собирайте фермы для рендеринга и многое другое ... с. 40



Cover illustration Radim Malinic

А также...

Постоянные рубрики

Обзоры



› ***FL Studio*** **(она же** ***Fruity Loops*****) наконец-то получила достойный Linux-эквивалент!**

Сравнение: календари



Linux Format Новости

ГЛАВНЫЕ НОВОСТИ: » Дешевые ноутбуки от различных производителей » Снова о патентах на ПО » Работа для Linux-специалиста » Государственный дистрибутив для школ » Mozilla Prism

» Рубрику ведет Илья Шпаньков



Школьный Linux: финишная прямая?



16 ноября закончился прием заявок на участие в конкурсе по созданию школьного Linux-дистрибутива, проводимом по инициативе Правительства РФ. В общей сложности побороться за тендер решили семь организаций и компаний:

» Компания «КОРУС Консалтинг» – заявленная стоимость проекта 5 млн. рублей;
» Томский государственный педагогический университет – заявленная стоимость проекта 70 млн. рублей;
» ЗАО «АйТи. Информационные технологии» – заявленная стоимость проекта 65 млн. рублей;
» ООО «Институт информационных технологий» – заявленная стоимость проекта 40 млн. рублей;
» ООО «РБК-Центр» – заявленная стоимость проекта 59,5 млн. рублей;
» ЗАО «Ланит» – заявленная стоимость проекта 63 млн. рублей;
» ЗАО «Вериселл Проекты» – заявленная стоимость проекта 66 млн. 495 тысяч рублей;

Один из представителей данного списка станет создателем школьного Linux-дистрибутива. По условиям конкурса, Правительство РФ готово выделить на данный проект 70 млн. рублей из федерального бюджета, и расчеты, представленные большинством соискателей, близки к данной цифре. Пожалуй, наибольший общественный резонанс вызвала стоимость проекта в 5 млн. рублей, заявленная компанией «КОРУС Консалтинг». В связи с этим в очередном пресс-релизе представители компании озвучили свою точку зрения на проект создания школьного Linux-дистрибутива. Как оказалось, на самом деле стоимость решения, предложенного специалистами «КОРУС Консалтинг», составляет около 60 млн. рублей, и в него входит разработка комплекта свободного программного обеспечения для образовательных учреждений, создание учебно-методической и справочной литературы, открытие специализированного информационного ресурса в Интернете, на котором можно будет отслеживать динамику подключения школ пилотных регионов к проекту, обеспечение ежедневной сервис-поддержки и обучение преподавателей. По словам же президента компании Игоря Широкова, «КОРУС Консалтинг» считает проект по созданию школьного Linux-дистрибутива делом государственной важности, а также новым перспективным направлением деятельности компании, поэтому и было принято решение использовать только небольшую часть бюджетных денег на реализацию проекта, а оставшиеся 55 млн. рублей инвестировать из собственных средств. Таким образом на сегодняшний день «КОРУС Консалтинг» оказалась единственной компанией, способной вкладывать собственные средства в развитие рынка свободного ПО в России.

Стоит также отметить, что один из кандидатов на право снабдить российские школы качественным свободным программным обеспечением, компания ALT Linux, уже поспешила анонсировать начало публичного бета-тестирования дистрибутива ALT Linux Junior 4, адаптированного к использованию в российских школах и дополненного специализированным ПО для сферы образования.





ПЕРЕТЯГИВАНИЕ каната по-европейски

Глядя на тот абсурд, что нередко творится в США в деле патентования программного обеспечения, европейские разработчики пока только улыбаются: твердое неприятие Европейской комиссией софтверных патентов создает благоприятную почву для дальнейшего развития свободного ПО. Между тем, не все так радужно, как это выглядит на первый взгляд. Дело в том, что вопросами патентования программного обеспечения на разных уровнях занимаются различные европейские организации, и не все из них относятся к вопросу патентов на программные продукты отрицательно. На сегодняшний день существуют три основные инстанции, способные как-то влиять на патентную систему, и силы в этой борьбе распределяются следующим образом:

» Законодательная власть в лице **European Patent Convention** выступает против патентования ПО;

» Исполнительная власть, представленная **European Patent Office**, всеми силами поддерживает такие патенты (что, в общем, логично – это их хлеб);

» Судебные власти, которые в данный момент единственные сталкиваются на практике с разрешением патентных споров на ниве ПО, действуют в соответствии с существующими европейскими законами и практически по всем патентным делам не признают патенты на ПО действительными.

Казалось бы, европейским разработчикам свободного ПО волноваться не о чем, но на самом деле ситуация может кардинально измениться уже в самое ближайшее время.



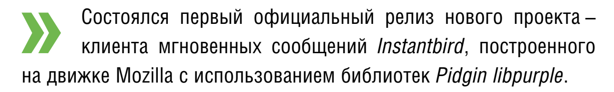
Дело в том, что определенные политические круги, активно поддерживаемые компаниями-разработчиками проприетарного ПО, сегодня пытаются изменить сложившуюся практику рассмотрения патентных судебных дел. Нововведения заключаются в том, чтобы рассматривать патентные тяжбы не в национальных судах, а силами специализированной судебной организации при Европейской комиссии. Пикантность же ситуации в том, что в соответствии с продвигаемыми предложениями состав судей данной организации будет практически полностью контролироваться **European Patent Office**. Таким образом, «правильно» подобранные судьи смогут кардинально изменить текущее положение дел, когда ни один патентный иск не решается положительно.

Скучать не приходится

Идея сверхдешевого компьютера, получившая широкую огласку благодаря проекту OLPC, оказалась довольно востребованной, что, в общем, легко объяснимо: людей небогатых в мире гораздо больше, чем тех, кому полторы-две тысячи долларов за персональный компьютер не кажутся неподъемными. Таким образом, в численном выражении рынок недорогих машин в разы больше сегмента «топовых» моделей, и этот факт заставил сразу несколько компаний выпустить на продажу соответствующие предложения, создав определенную конкуренцию ноутбуку XO, созданному в рамках проекта OLPC. Одним из наиболее разрекламированных вариантов стала настольная модель Everex TC2502 Green gPC, реализуемая через сеть супермаркетов Wal-Mart по цене $199. В качестве операционной системы используется дружественная к пользователю gOS (www.thinkgos.com), основанная на Ubuntu. Впрочем, данная цена не является конечной: в комплекте отсутствует монитор, за который придется выложить практически аналогичную сумму. Но и в этом случае полноценный компьютер стоимостью менее $500 выглядит вполне доступно.

Чуть более интересный вариант предложила компания ASUSTeK Computer Inc., выпустившая на рынок мини-ноутбук ASUS Eee PC 4G по цене от $299. Аббревиатура «Eee» расшифровывается, как «Easy to Learn, Easy to Work, Easy to Play» (Легко изучать, легко работать, легко играть), что, в общем, соответствует характеристикам ноутбука: в качестве операционной системы также используется один из вариантов Linux от компании Xandros, а аппаратная начинка Eee подошла бы и к более дорогим моделям. На прилавках США и Канады данное устройство появилось 1 ноября, а к концу года начнутся продажи в Европе и Юго-Восточной Азии.

Немного в другом направлении действует компания Intel, разработавшая свой вариант сверхдешевого ноутбука Classmate PC стоимостью около $200: это устройство в первую очередь предлагается для различных проектов, реализуемых в развивающихся странах. В частности, правительство Нигерии заключило с Intel контракт на поставку 17000 ноутбуков Classmate PC с предустановленной операционной системой Mandriva 2007 в рамках национального проекта по обеспечению образовательных учреждений компьютерной техникой. Также интерес к этому ноутбуку проявили правительства Мексики и Бразилии. Учитывая все вышесказанное, можно предположить, что проекту OLPC в ближайшем будущем придется побороться за место под солнцем.



КОРОТКОЙ СТРОКОЙ

» Состоялся первый официальный релиз нового проекта – клиента мгновенных сообщений *Instantbird*, построенного на движке Mozilla с использованием библиотек *Pidgin libpurple*.

» Проект Fedora Project официально объявил о выходе дистрибутива Fedora 8 (Werewolf), включающего такие новшества, как звуковой сервер *PulseAudio*, систему установки кодеков *Codec Buddy* и *IcedTea* – открытую среду Java на базе OpenJDK.

» Компания Mandriva объявила о том, что теперь участие в Mandriva Club – объединении пользователей, ранее существовавшем только на платной основе – будет свободным для всех желающих.

» Разработчики файлового менеджера *Far Manager*, весьма популярного в среде Windows, объявили о том, что с 26 октября все версии программы, начиная с Far 1.80 будут распространяться под модифицированной лицензией BSD.

» Компания Red Hat объявила о выходе обновления для Red Hat Enterprise Linux за номером 5.1 и сообщила о присоединении к проекту OpenJDK.

» Компания Novell начала в Китае продажи компьютеров с предустановленной операционной системой SLED 10.

» Вышла очередная версия интернет-телефона *Skype 2.0* для Linux, включающая поддержку обмена видеоданными.

Вольные хлеба сытнее

Не секрет, что проблема наличия достаточного количества квалифицированных Linux-специалистов является одной из наиболее актуальных. По этой причине все чаще можно прочитать объявления об открытии очередных учебных курсов по пользованию и управлению Linux-системами. Но, похоже, принципы свободы начинают проникать и в сферу подбора персонала. В частности, аналитическое агентство Foote Partners LLC провело исследование ситуации на рынке рабочих мест, требующих квалификации в области Linux, и получило неоднозначные результаты. Выяснилось, что число вакансий для Linux-специалистов выросло с прошлого года на 30%, что обусловлено ростом популярности данного ПО во всем мире. Также приверженцам свободного программного обеспечения будет приятно услышать, что отмечается стабильная тенденция к росту предлагаемых зарплат. Парадокс же в том, что одновременно с данными положительными моментами аналитики обнаружили, что спрос на сертифицированных специалистов в области Linux ПО постепенно снижается, и сегодня большинство компаний предпочитает приглашать на работу не тех, кто прошел официальное обучение работе с Linux, а людей без официального подтверждения квалификации. Возможно, объяснение кроется в том, что сертифицированные специалисты изначально обходятся дороже, но подобные повышенные расходы редко себя оправдывают: современное свободное ПО не требует каких-то особых навыков и может управляться обычными технарями среднего уровня.

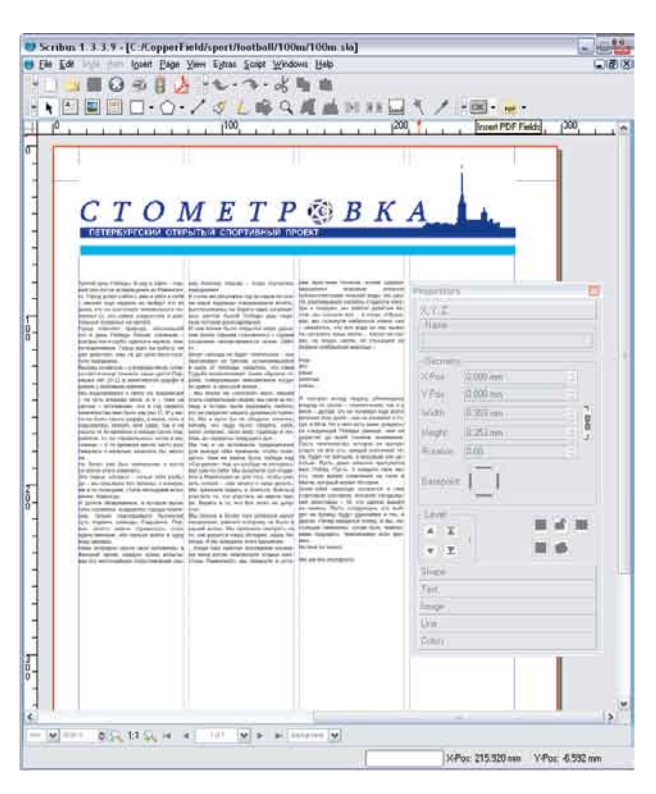


Мир через «призму» Mozilla

Медленное, но верное наступление эпохи Web 2.0 потихоньку изменяет привычные способы работы пользователя с прикладными программами: если раньше абсолютно все ПО устанавливалось стационарно на персональные компьютеры, то сейчас все больше онлайновых сервисов предлагают своим посетителям практически тот же функционал. В связи с этим активизировались и разработчики, специализирующиеся на привычном стационарном ПО, и одним из наиболее интересных и многообещающих примеров последнего времени можно назвать проект Prism, развиваемый фондом Mozilla. В двух словах – это система, позволяющая пользователю работать с web-приложениями практически так же, как и с обычными локальными. Запуск программ осуществляется обычным кликом на иконке приложения, размещенной на рабочем столе. Но это только внешняя сторона – как это часто бывает, выглядящее простым решение на самом деле содержит массу внутренних интересных особенностей. Так, к примеру, система Prism благодаря «хитрому» кэшированию данных допускает работу с web-приложениями в автономном режиме, без подключения к сети. Еще одно удобство, которого зачастую лишены пользователи онлайновых сервисов – полноценная поддержка технологии drag-and-drop, позволяющая легко управляться с передачей и сохранением данных на удаленном ресурсе. В общем, задумка довольно интересная, поэтому все желающие приобщиться к технологиям будущего приглашаются к тестированию бета-версии Prism. Установочные пакеты подготовлены для операционных систем семейства Mac OS X, Linux и Windows.

ОТКРЫТОЕ ПЕРВЕНСТВО:

на старте Стометровки



В Петербурге готовится к выпуску пилотный номер открытого издательского проекта.

Авторский спортивный проект «Стометровка», до этого 5 лет существовавший только в виде нерегулярных онлайн-публикаций, теперь станет полигоном сразу для нескольких экспериментов: во-первых, это попытка перенести OpenSource-принципы в сферу издательской деятельности, во-вторых – новое периодическое (как хотят надеяться организаторы) издание должно продемонстрировать возможности открытого программного обеспечения.

Именно поэтому подготовка макета новой «Стометровки», которая будет доступна в сети в виде сверстанной газеты в формате PDF, ведется в хорошо известных читателям пакетах *Scribus* (верстка), и *Inkscape* (графическое оформление). На старте в работе над выпуском номеров будут использоваться Windows-версии ПО, но со временем проект планируется полностью перевести на Linux-рельсы с вовлечением в производственный цикл таких инструментов, как *GIMP*, *Krita* и *Xara LX* (последние две программы заслуживают особого внимания: *Krita* существует только в мире Linux, а Windows-версия *Xara LX* – *XaraXtreme* – проприетарное коммерческое приложение).

«Стометровка» – это спортивно-аналитическое издание, в первую очередь ориентированное на петербургских любителей спорта – и в первую очередь футбола, но по мере развития проекта будет расширяться как целевая аудитория в географическом отношении, так и тематическое наполнение.

В планах издания – освещение событий из мира детского спорта, рубрики о фигурном катании, автоспорте, хоккее и других популярных видах спорта.

http://rodocop.pisem.net/100metroff.htm

» При подготовке данного выпуска новостей были использованы материалы сайтов: www.informationweek.com, www.fsfe.org, Enterprise Linux Log, www.pgz.economy.gov.ru, labs.mozilla.com, а также наших собственных корреспондентов.

LXF Репортаж

FOSS & Linux Solutions

В октябре этого года в Киеве была проведена первая в Украине выставка, посвященная Linux и свободному/открытому ПО. Параллельно с выставкой проходила конференция той же тематики. Впечатлениями от мероприятия делится **Тихон Тарнавский**.

Конференция OSDN (OpenSource Development Network) в Киеве стала уже – в полном смысле этого слова – традиционной. Стартовав осенью 2002 года, она с тех пор проводится с годичными интервалами вот уже шестой раз. На протяжении этого времени конференция пользовалась вниманием, начиная от индивидуальных разработчиков и просто энтузиастов свободного ПО – и заканчивая гигантами компьютерной индустрии: Sun, Novell, IBM и даже (кто бы мог подумать?) Microsoft.

В этом году нас ждало нечто большее, чем доклады о разработке и внедрении Linux-технологий и открытых стандартов. 25–27 октября в Торгово-промышленной палате Украины, расположенной в центре Киева, прошла первая выставка «FOSS & Linux Solutions», организованная Украинской Ассоциацией пользователей и разработчиков открытого и свободного ПО (UAFOSS).

Необходимость такого события в Украине назрела давно – это было видно по составу посетителей выставки и конференции. Если основным контингентом предыдущих мероприятий были «линуксоиды» со стажем и технические специалисты, то на этот раз людей пришло много и очень разных. Одновременно к нашему стенду подходят-интересуются директор частной компании и зав.кафедрой государственного университета. Рядом с ними – старшеклассник, который только начинает свой путь к Open Source и расспрашивает с нескрываемым интересом; и тут же человек почтенного возраста, ветеран войны, говорящий: «Да что вы мне рассказывать-то будете? Я в Linux шесть лет работаю. Я сам во всем разберусь.»

Стендов не очень много, но они тоже достаточно разнообразны по направленности. По краям той части, которая ближе к дверям в конференц-зал, находились стенды спонсоров выставки: компании «Media Magic», занимающейся автоматизацией предприятий, и крупного издательского дома компьютерной тематики «Комиздат». Другие хорошо известные спонсоры мероприятия – Sun и Novell – стендов своих не организовывали; ограничились докладами на конференции, хотя флаг SUSE на стенде «Media Magic» висел.

Между ними, как бы окружая выход из зала, расположились стенды ALT Linux, LinuxCenter/LinuxFormat, Ubuntu и поставщика компьютерной техники «Комсервис». В другой половине зала были размещены стенды «Компьютерной группы «Дрогобыч» (дистрибьютор и внедренец Etersoft в Украине), компании «Юстар» (поставщик высоконадежных информационных систем), а также самый большой стенд самой Ассоциации и – примкнувший к нему – интернет-журнала [Root@UA].

Выставка по времени практически совпала с началом деятельности LinuxCenter в Украине. Поэтому к стенду был большой интерес, и, похвалив любимый журнал, отметим, что многих особенно радовала появившаяся возможность покупать и выписывать в Украине LinuxFormat. Общее настроение по этому поводу можно охарактеризовать одной фразой, которую нам доводилось слышать чаще всего: «Ну наконец-то!»



› **А вот и посетители.**





› **Только открылись...**

К стенду ALT Linux посетителей буквально притягивало обаяние главы компании Алексея Смирнова, который активно общался со всеми интересующимися.

По посещаемости стенда Ubuntu тоже было видно, насколько популярен в Украине этот дистрибутив. Что происходило в другой половине зала, я, к сожалению, не видел, т.к. все три дня не мог отойти от стенда.

Конференция пользовалась очень большой популярностью. Первые два дня на выставке, которая на перерывах была буквально забита посетителями, во время докладов движение происходило несколько вяло. В субботу ситуация изменилась: желающих уже хватало с лихвой и на два зала, где проходили доклады, и на то, чтобы не давать скучать участникам выставки на своих стендах.

В целом можно сказать: старт дан очень хороший, первая выставка показала, что в Украине уже есть большой интерес к Linux и свободному ПО, и, думается, этот интерес будет только нарастать.

В дополнение к впечатлениям от выставки я пообщался с координатором организовавшей ее Ассоциации – Юрием Радченко.

LXF: **Эта выставка первая, но в дальнейшем, я думаю, планируется проводить ее ежегодно?**

Юрий Радченко (ЮР): Да, конечно. »



› Выступает А.Смирнов; в президиуме — координатор UAFOSS Ю.Радченко.

» **LXF: Почему именно в этом году была подготовлена первая выставка? Когда и как Ассоциация пришла к этой идее?**

LXF: Конференция OSDN проводится уже шестой год кряду. Однако формат профессиональной конференции — не совсем то, что необходимо на данном этапе развития общей тематики свободного/открытого ПО. Уже с 2005 года было понятно, что мероприятие нужно выводить на более массовый уровень — ибо основная проблема на Украине — это отсутствие широкой информации о возможностях, да и о самом факте существования, свободной/открытой интеллектуальной собственности.

LXF: В этот раз участников было не очень много, выставка получилась скорее дополнением к конференции. Планируете ли вы в дальнейшем выводить выставку на новый уровень?

ЮР: Во-первых, я бы не сказал, что было мало участников. Согласно нашей регистрации, мероприятие посетили около 800 человек: до 200 представителей от государственных, учебных, научных и общественных организаций в первом потоке, и около 600 человек – за два дня «профессиональной» части, которую пришлось разделить на два потока из-за большого числа докладов. Сравните: разрекламированная в «Компьютерном обозрении» конференция «Корпоративная автоматизация» собрала 50 человек, HPC Day-2007 – 150...

LXF: Вы правы, количество посетителей производило впечатление. Но я не совсем это имел в виду: под участниками я подразумевал не посетителей, а экспонентов выставки.

ЮР: Выставка в этом году была скорее иллюстративной. Но общая идея состоит в том, повторюсь, чтобы вывести все мероприятие на действительно масштабный уровень. Учитывая, что на прошлой конференции было около 150 человек – это, я полагаю, задача посильная. Скорость же ее выполнения зависит от того, как быстро руководители фирм поймут, что СОПО действительно является международным трендом развития...

Касательно нового уровня – да, наша идея – за несколько итераций довести мероприятие до уровня крупного и основного для страны ежегодного форума по вопросам разработки и внедрения новых типов интеллектуальной собственности, особенно – свободного/открытого ПО.

LXF: По моему впечатлению, большой процент посетителей выставки — так называемые «домашние пользователи», которых вопросы разработки и внедрения интересуют не в первую очередь. Что, на ваш взгляд, эти люди могли почерпнуть для себя на этой выставке?

ЮР: То, что, думаю, и почерпнули – наглядное доказательство возможности установки свободного ПО на домашнюю систему, со всеми привходящими аспектами – от наличия удобного делового ПО до игр с поддержкой современных трехмерных интерфейсов.

LXF: Будет ли выставка впоследствии тоже совмещена с конференцией, или может стать самостоятельным событием?

ЮР: Думаю, будет.

LXF: Внимание каких компаний вы планируете в дальнейшем привлечь к этой выставке? На какого посетителя собираетесь ее ориентировать?

ЮР: «Вход» на мероприятие открыт любым компаниям, которые понимают, что производство и распространение свободной/открытой интеллектуальной собственности – это мировой тренд. Что касается какого-то «особого посетителя», на которого обязательно нужно ориентировать любое мероприятие – то это элементы западного подхода к маркетингу; по нашему мнению, здесь западный маркетинг не работает.

LXF: Другими словами, организаторы ориентируют выставку не на какие-либо целевые ниши как посетителей, так и экспонентов, а на всех и каждого, кому будет интересна ее тематика?

LXF: Одно не исключает другого. В этом году в отдельный поток выделилась образовательная тематика; первый день был посвящен аудитории из госсектора. Это и есть позиционирование. Но, с другой стороны, основная слабость новых экономических моделей интеллектуальной собственности на Украине – это то, что о них известно очень малому проценту населения. Вот в этом смысле нас и интересует максимальное расширение мероприятия, превращение его в общенациональное.

LXF: Спасибо вам за интервью, и за интересную выставку!



› На конференции.

LXF Репортаж

Freecode приходит в Россию

В конце октября в Москве прошла церемония открытия первого в России офиса норвежской компании Freecode International. Что это значит для отечественного Linux-сообщества? Разбирается **Григорий Рудницкий**.

Распространение свободного программного обеспечения на российском корпоративном рынке сдерживается, среди прочего, недостатком технической поддержки, а также услуг в области внедрения и консалтинга. В недалеком будущем ситуация может измениться, так как в Россию пришла норвежская компания Freecode, которая и будет заниматься всем вышесказанным. Пока офис Freecode International открыт в Москве, но скоро появятся ее представительства и в других городах. Специально для ознакомления российских заказчиков со свободными программными продуктами нашу страну посетил генеральный директор Freecode International Гейр Исене [Geir Isene]. Он постарался раскрыть возможности и преимущества Open Source не ИТ-специалистам, которые и так неплохо с ними знакомы, а управляющим менеджерам – генеральным и коммерческим директорам. Мы задали ему несколько вопросов.

LXF: **Каковы стратегические цели прихода вашей компании в Россию?**

Гейр Исене (ГИ): Главная причина состоит в том, что Россия – это огромный быстрорастущий рынок. Это даже лучший рынок для развития Linux, чем Западная Европа. Возможно, российские клиенты не очень счастливы по причине того, что им приходится платить деньги американским компаниям.

LXF: **Сегодня денежные потоки на ИТ-рынке идут во вполне объяснимом направлении — в сторону Microsoft и прочих поставщиков проприетарного ПО. Собираетесь ли вы изменить расстановку сил на ИТ-рынке и перенаправить денежные потоки на нем?**

ГИ: Очень хороший вопрос. С моей точки зрения, российские деньги должны быть реинвестированы именно в Россию. Все, что я хочу сделать – это стать катализатором данного процесса. Наша компания видит свою роль в том, чтобы стать катализатором для распространения и развития российских разработчиков и поставщиков свободного ПО. Эти разработчики должны предоставлять реально нужные для российского бизнеса решения. Нам нужен русский Linux для русского бизнеса, который мог бы использоваться в любых задачах.

LXF: **У свободного ПО есть еще ряд проблем. К примеру, проблема с драйверами, поддержкой оборудования. Как ваша компания может повлиять на ее решение?**

ГИ: Вы должны понимать, что подобные проблемы случаются с и коммерческим ПО. Кроме того, мы работаем в тесном сотрудничестве с компанией Canonical. Это сотрудничество как раз и должно помочь решению таких проблем. Каждый раз, когда Canonical сталкивается с какой-либо проблемой у своего клиента, к примеру, с отсутствием поддержки того или иного оборудования, приходится даже писать отсутствующие драйверы и передавать их свободному сообществу.

Приведу пример. Мы столкнулись однажды с тем, что в одном из серверов компании HP возникла проблема с Linux-драйвером для RAID-контроллера. Мы связались с разработчиком этого драйвера, который является штатным сотрудником хьюстонского офиса HP, и помогли ему решить данную проблему.

LXF: **Вы сейчас будете активно популяризировать свободное ПО для российских заказчиков. Какие действия вы планируете предпринять в этой области?**

ГИ: План простой – открыть больше офисов. Нам нужно примерно 20 офисов по всей России. Мы планируем открыть их в течение ближайших трех лет.

LXF: **Сколько сотрудников вы планируете набрать, и в каких областях в первую очередь требуются специалисты?**

ГИ: Действительно, нам очень нужны люди! Нужны программисты, системные администраторы, способные поддерживать не только Linux, но и BSD-системы. Также нам очень необходимы менеджеры проектов и консультанты, способные реализовывать методику ITIL. И, конечно, нам нужны люди, способные делать неординарные вещи, чтобы распространять идею свободного ПО в массы. Фактически, нам нужно по 10 сотрудников в каждом офисе. Лучше иметь 10 сотрудников в одном офисе, а всего таких офисов – 20, чем один офис с 200 сотрудников.

LXF: **В России есть много компаний, которые занимаются разработкой свободного ПО. Вы активно сотрудничаете с компанией Canonical. Готовы ли вы к сотрудничеству с российскими Linux-разработчиками?**



› Гейр Исене

ГИ: Для нас очень важно сотрудничество с российскими компаниями. Мы стремимся объединить всех в стремлении к общей цели – предоставлять хороший продукт конечному пользователю.

LXF: **И последний вопрос. Во многих странах государственные структуры активно рассматривают свободное ПО в качестве альтернативы продукции Microsoft. Планирует ли ваша компания работать в этом направлении с российскими госструктурами?**

ГИ: Очень интересный вопрос. Мы работаем с норвежским правительством, для них мы подготовили каталог свободного ПО. Хорошо было бы, чтобы и российский офис занимался тем же самым в России. Основная причина, по которой это следует делать, состоит в том, что мы заинтересованы, чтобы правительство и госструктуры получили то решение, тот продукт, от которого они выиграют и который будет способствовать их развитию. Для нас главное – не просто делать бизнес, но делать правильные вещи, чтобы они помогали людям. А деньги и так придут. LXF

LINUX FORMAT

Обзоры



Новинки программного и аппаратного обеспечения в описании наших экспертов



Алексей Федорчук
Его слабости – mass storage, разметка диска и файловые системы.

Linux и творческая интеллигенция

В последнее время много говорится о Linux в образовании, что, конечно, очень благородно. Но стоит взглянуть на этот вопрос с другой стороны: а что используют инженеры человеческих душ, на произведениях которых, когда они станут классиками, будет учиться подрастающее поколение?

Будучи модератором на нескольких литературных форумах, я часто задавался вопросом: каковы побуждения поэтов и писателей к использованию Windows и ее приложений, в первую очередь, *MS Office*, из которого реально применяется, естественно, только *Word*?

Ведь, казалось бы, уж кому-кому, а поэтам и писателям сам Бог велел использовать любой развитый текстовый редактор – и, разумеется, на любой платформе, позволяющей создавать и обрабатывать тексты и передавать их по назначению – в редакции ли бумажных издательств или на сетевые ресурсы.

Опрос среди лично и виртуально хорошо знакомых мне литераторов (в данное понятие я включаю также литературоведов, критиков, публицистов и литературных переводчиков) выявил две причины. Первая – банальная привычка. Но она ведь не столь уж стара – мои ровесники помнят времена, когда писали даже не авторучками, а ручками, опускаемыми в чернильницы. А до того – так вообще гусиными перьями. Кстати, существует мнение, что именно гусиными перьями создавались вечные произведения. А перьями вечными – произведения гусиные.

Вторая же, и, как мне кажется, главная причина – это просто незнание возможностей обработки текста, которые предоставляют связка из текстового редактора и полудюжины классических Unix-утилит. Так не помочь ли братьям-литераторам узнать о них? Это высвободит им время на сочинение тех самых вечных произведений, которые будут читать наши дети и внуки...

alv@posix.ru

Сегодня мы рассматриваем...

QNAP с. 12



› **Тихий, маленький и емкий — и его не стыдно поставить на видное место!**

LMMS с. 13

› **Сваять адски-супер-пупер-модный микс на этой дьявольской машине — плевое дело!**

НАШ ВЕРДИКТ: пояснение

Все попавшие в обзор продукты оцениваются по одиннадцатибальной шкале (10 – высшая оценка, 0 – низшая). Как правило, мы оцениваем функциональность, производительность, простоту использования и цену, а для бесплатных программ учитывается документация. Кроме того, мы всегда выставляем общую оценку, демонстрирующую наше отношение к продукту.

Выдающиеся решения могут получить престижную награду «Top Stuff». Номинантами становятся лучшие из лучших – просто высокой оценки здесь недостаточно.

LINUX FORMAT TOP STUFF

Рассматривая свободное ПО, мы обычно указываем предпочтительный дистрибутив. Иногда это означает компиляцию из исходных текстов, но, если разработчики рекомендуют Autopackage, мы следуем этому совету.



Вердикт

Google Earth

Разработчик: Google
Сайт: http://earth.google.com
Цена: Бесплатно по закрытой лицензии

Функциональность	10/10
Производительность	9/10
Простота использования	9/10
Документация	9/10

» *Если весь мир – сцена, то Google Earth – театр. Простая в использовании, захватывающая и ободряюще практичная программа.*

Рейтинг 9/10

Professor Fizzwizzle and the Molten Mystery

А вот **Грэм Моррисон** считает, что нет здесь никакой мистики – просто хорошая игра. Забудьте о 3D-графике и встроенных клипах: все дело в правильной организации уровней.

Вкратце...

» Платформенная головоломка, напоминающая Sokoban, только вид «сбоку». Среди подобных – свободные версии *Sokoban* или *Fish Fillets*.

Пора вспомнить, что 3D-графика – не единственный способ создания компьютерных игр. Многие новейшие игрушки для Nintendo DS полностью подтверждают это, а ряды любителей достопочтенного *Тетриса* за 20 лет почти не поредели: что-то в этих играх цепляет за душу. Нельзя сказать, что уважаемый профессор дурно одет, но мультяшная двухмерная графика и логическая природа игрового процесса – определенно возврат к золотому веку компьютерных игр. Это совсем не плохо, особенно для бедного играми мира Linux. Но жанр двумерной логической игры среди разработчиков открытого ПО весьма популярен – достаточно взглянуть на игровое меню вашего дистрибутива, чтобы убедиться в этом. Это ужесточает конкуренцию, особенно для коммерческих проектов.

Простак-профессор

Игра во многом напоминает *Sokoban* – известную японскую головоломку, в которой герой передвигает ящики по полу заводского цеха. Только вместо вида сверху, как в *Sokoban*, мир нашего профессора виден сбоку, в традициях старых платформенных игр. Приятные мультяшные картинки проводят игрока сквозь нехитрый сюжет, сводящийся к погоне за экс-ученицей профессора и ее армией летучих ботов. Честно говоря, вникать мы не стали, а сразу же ринулись в бой. На начальных уровнях учебные сценки знакомят с механикой игры. Профессору понадобится двигать ящики и бочки, пробираться сквозь ворота, через заборы и по пересеченке: на пути ему встретится немало преград. Организация уровней тщательно продумана. Каждый новый экран заставляет на несколько мгновений задуматься над выбором наикратчайшего пути для профессора. При неудаче можно нажать U, чтобы отменить последнее действие, или R для немедленного перехода на начало уровня. Игра затягивает и увлекает – создатели приложили к этому немало умственных усилий. Закончив прохождение уровня, можно даже сравнить свое достижение с онлайн-турнирной таблицей. В ней-то вы и обнаружите, что ваши способности к решению проблем на сотни мест отстают от показателей лидеров. Как тут не нажать R и не сократить свое прохождение еще на пару движений?!



› Профессору нужно попасть из одного пункта в другой и не получить при этом по голове. Более высокие уровни игры еще подливают масла в огонь.

Другой уровень

В игре более 210 уровней, все они разбиты на три категории трудности. Простейший из них предназначен для детей. Прекрасно задумано: красочная графика, увлекательный игровой процесс и даже дурашливый сюжет доставят вашим отпрыскам море удовольствия. У игры есть даже активное сообщество пользователей, создающих собственные уровни с помощью встроенного редактора – вот вам и занятие для маленьких соперников. Именно внимание к деталям отличает эту игру от большинства подобных головоломок из мира открытого ПО. И именно это отличие стоит запрашиваемой цены, особенно если у вас есть дети, для которых главный недостаток Linux-компьютера заключается в нехватке игр. LXF



Свойства навскидку



Детский уровень

Треть уровней игры задуман так, чтобы заинтересовать и увлечь маленьких детей.



Онлайн-соперничество

Сравнив свои успехи с интегрированной онлайн-турнирной таблицей, невольно хочется повторить игру.

LINUX FORMAT Вердикт

Professor Fizzwizzle and the Molten Mystery

Разработчик: Grubby Games
Сайт: http://grubbygames.com
Цена: ок. $20

Сюжет	7/10
Графика	9/10
Увлекательность	7/10
Цена	8/10

» *При 60-дневной гарантии возврата денег вы ничего не теряете, взяв игру на пробу!*

Рейтинг 8/10

QNAP TS-109 Pro

Грэм Моррисон рассматривает компактный сетевой накопитель, который не испортит своим присутствием ни серверную, ни гостиную.

Вкратце...

» Встроенный Linux с большим жестким диском. Содержит все известные серверы, а также порты для USB-устройств и принтеров.

СПЕЦИФИКАЦИЯ
CPU: 500-МГц интегрированная система ARM926EJ-Sid.
RAM: 128 МБ DDR II
Порты: Gigabit Ethernet, 3 USB-порта.
Накопитель: 500 ГБ SATA-диск Seagate Barracuda

Сетевые накопители (Network attached storage, NAS) сегодня нарасхват. Наши ноутбуки, настольные компьютеры, PDA и игровые приставки бесконечно требуют свежих данных. NAS – идеальное хранилище для этих данных, ведь любое устройство, способное войти в сеть, может подключиться и к NAS. Посмотрите на спецификацию (слева): конфигурация больше напоминает персональный компьютер, чем винчестер с Ethernet-портом. Небольшой, но прочный алюминиевый корпус вмещает только один жесткий диск, поэтому RAID «отдыхает»; наш экземпляр был оснащен 500-ГБ приводом Seagate Barracuda. Сложив все это, получаем довольно мощное устройство. А главное, оно работает под Linux. Включим его и подождем примерно 30 секунд для загрузки (сопровождаемой парой гудков и шоу лампочек на передней панели) – и можно входить в QNAP без дальнейшей возни с настройками. Немногие устройства способны на такое. Версия ядра свежая (заявлена как *Linux qnap 2.6.12.6-arm1*), сам дистрибутив – модифицированная версия BusyBox v1.01, а для пытливых в запасе имеется ARM-версия Debian Etch.



› Крохотные 21 x 6 x 18 см – просто смех на фоне среднего Linux-компьютера.

Web-интерфейс

Большинству пользователей не придется даже притрагиваться к командной строке. 109 Pro предлагает на выбор любой сервер данных, который только есть в мире открытого ПО, включая *Samba*, *AppleTalk*, *NFS*, *Apache* (с PHP), FTP, *MySQL*, iTunes и даже *BitTorrent*-клиент. И все это управляется через удобный web-интерфейс. Например, чтобы сконфигурировать устройство как разделяемый ресурс Windows, достаточно выбрать в меню Administration (Администрирование) пункт Microsoft Networking (Сети Microsoft), ввести название вашей рабочей группы – и вот вы уже читаете/пишете в Public-разделе накопителя. Доступ к нему может контролироваться на уровне пользователей через другую административную панель, что значительно проще настройки и использования типовой инсталляции *Samba*. Несколько разделяемых ресурсов настроены заранее. **Public** – раздел общего назначения, а в папке **Qdownload** хранятся FTP- и *BitTorrent*-файлы. *QWeb* – это файлы, обслуживаемые web-сервером; USB-накопитель доступен через **Qusb**. Наконец, папка **Qmultimedia** предназначена для хранения видео, музыки и фотографий. Открыв эту папку с главной страницы web-интерфейса, можно просмотреть содержание фотоколлекции в виде миниатюр. Правда, процесс создания этих миниатюр недопустимо долог – сканирование каких-то 18 изображений занимает несколько минут.

› Мультимедиа-компонент создаст web-альбом из вашей фотоколлекции. Запаситесь терпением: придется подождать.

Потоковые медиа

Медиаконтент может транслироваться клиентам сети с помощью DNLA-совместимого сервера *TwonkyMedia 4.4* (мы рассматривали его пару месяцев назад, в **LXF96**). Это позволяет слушать музыку, просматривать фото и смотреть фильмы на самых разных устройствах, от Playstation 3 и XBox 360 до PDA. Есть на устройстве и альтернатива, *Slimserver* от Logitech. Это одно из лучших решений для потокового вещания звука, и вовсе не обязательно иметь дорогостоящий *Squeezebox*, чтобы извлечь из него максимум пользы. Прослушивать музыку можно с помощью любого браузера через web-интерфейс. Как вы уже заметили, у QNAP 109 Pro столько функций, что пользователи обычных NAS будут просто обескуражены. Вероятно, именно поэтому в названии возникла приставка 'Pro'. Ну, а того, кто привык работать с Linux-серверами, всем этим не удивишь. QNAP столь мощный набор функциональности к лицу. Устройство тихое, безотказное, а его внешний вид полностью соответствует современным представлениям о цифровой аппаратуре. Единственное, что нас слегка царапнуло – почему Linux-разработчикам не выразили особой благодарности за все то великолепие, что они предоставили бесплатно? LXF



QNAP TS-109 PRO
Разработчик: QNAP
Сайт: www.qnap.com
Цена: Без диска: ок. $380 (с НДС)
с 500 ГБ-диском: $580 (с НДС)

Функциональность	10/10
Производительность	8/10
Удобство использования	9/10
Документация	8/10

» *Коллекция лучших Linux-серверов на солидном винчестере.*

Рейтинг 9/10





LMMS 0.3.0

Грэм Моррисон покорен достижениями лучшей Linux-программы для создания музыки. Каждому – срочно по TB-303!

Вкратце...

» Быстродействующая программа для создания музыки с эффектами реального времени, возможностями управления и синтеза. Можно попробовать также *EnergyXT* или *Rosegarden*.

Гвоздь LXF74, *LMMS*, ныне стала стандартным Linux-инструментом созидания для техно, электро-, транс- DJ и музыкантов. В *LMMS* все подчинено одной мысли: писать музыку должно быть легко и приятно. Теперь программу вполне можно сравнить с ее прообразом – Windows-программой *FL Studio* (известной ранее как *Fruity Loops*). Оба приложения обладают пользовательскими интерфейсами образца Amiga. На раздвижной панели модулей в левой части окна композиции содержится весь музыкальный контент: синтезаторы, брейкбиты и сэмплы. Тут есть и множество образцов музыки, созданной другими людьми – свидетельство незаурядных возможностей *LMMS*.

Время творить

Как и в реальном мире, мелодии создаются из блоков, содержащих ноты. Блок может быть четырьмя тактами мелодии, тактом, содержащим метр, или даже единственной нотой, используемой для проигрывания отрезка заранее записанного аудио. Блоки создаются с помощью редакторов Beat+Bassline или Piano Roll. Редактор Beat+Bassline больше подходит для создания ритмов – он работает как простой триггер, проигрывая звук всякий раз при активации одной из 16 доступных кнопок. Например, можно перетащить сэмпл басового барабана в редактор Beat+Bassline и щелкнуть на первой, пятой, девятой и тринадцатой кнопках для проигрывания сэмпла на первой доле каждого такта. Попробуйте проделать это самостоятельно и зафиксируйте сэмпл – вот вы и сочинили свой первый ритм. Быстро, легко, и звучит замечательно. Два маленьких дополнения показались нам особенно приятными: групповой контроль громкости инструментов в редакторе Beat+Bassline (например, можно уменьшить громкость всей ударной установки единственным контроллером), и слайдер для «размывания» четкого метрономического ритма. Ну, а если вам нужен более полный контроль над запуском каждой ноты, можно воспользоваться редактором Pattern (редактором образцов). Он использует матричное представление для создания ноты определенной высоты и длительности – можно даже создать запись, а затем проиграть «вживую» на компьютерной или MIDI-клавиатуре (если таковая имеется). Затем различные блоки нот размещаются в указанном вами порядке в окне Song Construction (Композиции).

Замечательно, что в *LMMS* вы не прикованы к одним сэмплам: по умолчанию включено десять синтезаторов реального времени, открывающих бесконечные возможности расширения и контроля звуковой палитры. Здесь можно воссоздать и кваканье знаменитой Roland'овской TB-303; есть 3 осцилляторных синтезатора, которым ничего не стоит издать любой доступный синтезатору звук. Есть аддитивный синтезатор, струнный синтезатор, таблично-волновой синтезатор и даже звуковой генератор, использующий речевой движок Festival. А лучше всего – интеграция VST. VST-инструменты – это стандарт для программных синтезаторов на Windows, но не свободный стандарт, и его включение в открытое ПО часто исключало распространение бинарных файлов. Команда *LMMS* сделала невозможное: произвела реинжиниринг VST-спецификации, и теперь она включается в *LMMS* бесплатно, под свободной лицензией. А работает программа изумительно. Мы загрузили несколько общедоступных VST-инструментов, открыли их библиотеки в модуле *LMMS* VST и получили совершенно новые источники звука. Рассказывать об *LMMS* можно долго, но лучше просто взять и попробовать самому. Все-таки, главное достоинство программы – простота и непринужденность музыкального творчества. LXF



» **Кривая обучения интерфейсу довольно крута, но по мере освоения материала становится более пологой.**

LMMS: основные функции



Автоматизация Любой орган управления можно автоматизировать, просто щелкнув на нем правой кнопкой мыши.

Программные синтезаторы LMMS обладает собственными синтезаторами изэффектами.

Обработка Работайте с сэмплами: меняйте форму волны, управляйте высотой, уровнем звука и т.п.

Контроль фильтров и огибающей Фильтруйте источники и управляйте их динамикой.

Матричный редактор Просто нарисуйте ноту, которую хотите услышать.

Beat/Bassline Щелчок в этой области управляет барабанными битами и отдельными нотами.

Композиция Составьте мелодии и закрепите блоки, чтобы освободить ресурсы.

Стойка с эффектами Каждый инструмент оснащен различными эффектами, включая встроенные и LADSPA.

LINUX FORMAT Вердикт

LMMS 0.3.0

Разработчик: Ансамбль LMMS
Сайт: http://lmms.sourceforge.net
Цена: Бесплатно под GPL

Функциональность	10/10
Производительность	10/10
Простота использования	8/10
Документация	8/10

» *Неплохо бы включить синхронизацию эффектов в масштабах проекта, да добавить четкости интерфейсу – все остальное безупречно.*

Рейтинг 9/10

SmoothWall Express 3.0



Оживите старый ПК, заодно оградив свою сеть от «плохих парней» из Интернета. **Нейл Ботвик** пробует новейшую версию классического дистрибутива-брандмауэра.

Вкратце...

» Превращает старый (ну, или не очень старый) компьютер в полноценный брандмауэр, маршрутизатор, прокси-сервер и интернет-шлюз.

См. также: *IPCop* или *ClarkConnect*.

DMZ

» Демилитаризованая зона, или DMZ – часть сети, доступная внешнему миру. Это способ уберечь основную часть сети от внешних вторжений, имея при этом серверы, подключенные к Интернету. Доступ машин, находящихся в DMZ, к остальной части локальной сети тоже ограничен, чем обеспечивается защита сетевых компьютеров даже при поражении сервера.

Резонно спросить: «В ядре есть *iptables*, графических оболочек для брандмауэров – пруд пруди, зачем же еще отдельная машина?». Но ведь при запуске брандмауэра на машине, которую надо защитить, трафик достигает компьютера раньше, чем брандмауэр может остановить его – это все равно что закрывать межкомнатные двери, оставляя входную дверь открытой. Другое крупное преимущество отдельного брандмауэра – на нем нет лишнего ПО: настольный компьютер несет на себе сотни полезных и нужных программ, а на брандмауэре устанавливается то, что необходимо для выполнения одной-единственной задачи. Сообщество SmoothWall поможет расширить диапазон полезных функций. В большинстве случаев для брандмауэра бывает достаточно Pentium 200 с 128 МБ ОЗУ, хотя для работы большого количества служб в загруженной сети конфигурацию можно усилить (особенно память).

SmoothWall Express задействует от двух до четырех интерфейсов. Простейший вариант, понятно, два: Red (Красный), для подключения к Интернету, и Green (Зеленый), обращенный к локальной сети. Красным интерфейсом может служить обычный модем (пользователи могут подключаться и отключаться через администраторскую страницу SmoothWall), модем SpeedTouch USB ADSL или Ethernet-подключение. Все остальные интерфейсы – Ethernet, из которых два дополнительных: Orange (Оранжевый) – для DMZ (см. врезку слева), и Purple (Лиловый) – для беспроводной сети. Последний вариант для SmoothWall 3.0 новый, он позволяет накладывать строгие ограничения на беспроводные подключения. Неплохое дополнение, учитывая повышенную подверженность беспроводных сетей атакам извне.

» У web-администратора все под контролем, от настройки до учета трафика в реальном времени.

Инсталляция проста: вставьте диск и следуйте подсказкам. Заранее надо знать только тип «красного» интерфейса и общее количество интерфейсов, что определяется конфигурацией конкретной сети. Установив SmoothWall Express, можно убирать диск, монитор и клавиатуру: все дальнейшее администрирование осуществляется через web-интерфейс с любого сетевого компьютера (с условием, что пользователь компьютера знает пароль администратора).

Браузер-администратор

Web-интерфейс хорошо продуман, руководства дополнены онлайн-документацией. Бывает, что нужный параметр находится не с первого раза, но ведь параметров-то – море. В целом, интерфейс обладает и полнотой, и простотой в использовании. Крутой хакер с аллергией на мышей может воспользоваться для управления системой SSH, но в этом случае возрастает риск «сделать что-то не так».

SmoothWall Express снабжен всем, что можно ожидать от брандмауэра/маршрутизатора – DHCP, NAT, пробросом портов, выборочной блокировкой/допуском IP-адресов, и многими другими функциями. Дополнения включают прокси-серверы для web, IM, SIP и POP3, в последний встроена антивирусная проверка с *ClamAV*. Есть возможность управления трафиком, VPN, NTP-сервер и детальные отчеты, плюс графическое отображение трафика в реальном времени.

В настройке брандмауэра важно не ошибиться. Неверно настроенный брандмауэр хуже, чем его полное отсутствие: в последнем случае вы хотя бы осознаете свою уязвимость! В составе программы два PDF-руководства, 40-страничное руководство по инсталляции и 86-страничное руководство администратора, содержащие все сведения, которые только могут понадобиться для сборки безопасного брандмауэра/маршрутизатора в большинстве ситуаций, а также предупреждения о том, чего делать не стоит. Все это подкрепляется онлайновой справочной системой, доступной через web-интерфейс. LXF

Что нового в 3.0

Полный список слишком велик, перечислим лишь хиты:

- » Беспроводная сеть.
- » 64-битная версия.
- » Ядро 2.6.
- » Загрузка и инсталляция обновлений одним щелчком мыши.
- » Новое графическое отображение трафика.
- » Поадресная статистика трафика.
- » IM- и SIP-прокси.
- » Временной контроль доступа.
- » Фильтрация исходящего трафика.
- » Поддержка QoS для управления и формирования трафика.
- » Новая привлекательная тема: вернулся белый медведь!
- » Возможность установки коммерческих пакетов.



SmoothWall Express 3.0

Разработчик: SmoothWall Open Source Project
Сайт: www.smoothwall.org
Цена: Бесплатно под GPL

Функциональность	9/10
Производительность	8/10
Простота использования	8/10
Документация	9/10

» *Красивый, надежный и мощный дистрибутив, пригодный для большинства локальных сетей. Крупный шаг вперед по сравнению с версией 2.0.*

Рейтинг 9/10

Сравнение



Каждый месяц мы сравниваем тонны ПО – а вы можете отдыхать!

Календари



Вам, поди, и невдомек, что выбор программ-календарей весьма богат. **Грэм Моррисон** исследует варианты.



Paul Blachford

Про наш тест...

Календарь – явно не главная программа на вашем рабочем столе Linux, поэтому в первую очередь нас интересовали удобство использования и простота. Без них вряд ли кто откроет подобную программу дважды, не говоря уж о систематическом ведении ежедневника. Добавление и редактирование событий должно быть легким, а интерфейс – понятным и функциональным. Кроме того, мы учитывали интеграцию с другими инструментами планирования (например, списком контактов или почтовым клиентом), хотя для удобной и гибкой программы это не обязательно. А еще хотелось найти приложение, способное управляться с нашими учетными записями *Google Calendar*, а также импортировать и экспортировать данные в других известных форматах.

Наш выбор

Понимаем: первая ваша реакция – да с чего тут суетиться? Календарь – не более чем повседневный инструмент: бросаешь на него мимолетный взгляд утром, испускаешь вздох при виде громадья планов, закрываешь и забываешь. Но ведь календари способны на большее – они могут здорово облегчить вашу жизнь. К тому же, разные приложения весьма непохожи друг на друга. Не у всех нас, как у Кондолизы Райс, каждая секунда на учете, но разве лишним будет напомнить о скором мамином дне рождения? Настольных собратьев изрядно потеснили онлайн-календари, типа *Calendar* от Google: они удобнее тем, что позволяют создавать несколько календарей и обмениваться ими с другими пользователями, а доступ к ним открыт с любого компьютера, подключенного к Интернету – неоценимые преимущества для тех, кто под предлогом сверхзанятости норовит уклониться от участия в семейном сборище. Но популярность онлайн-календарей заставила настольные программы немедля адаптироваться.

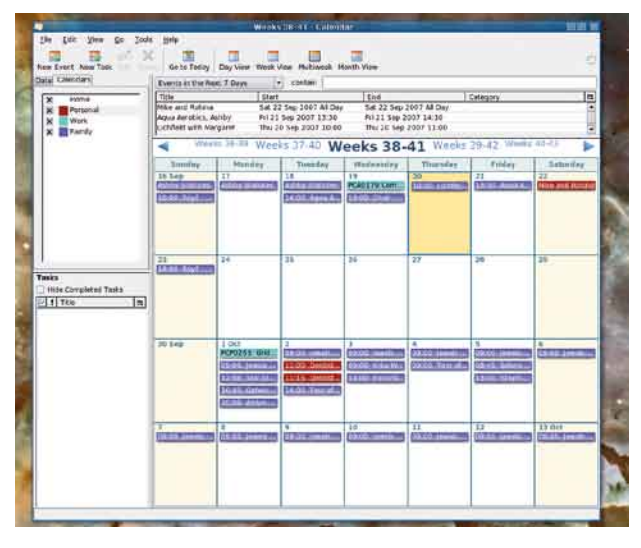
Равнение на Apple

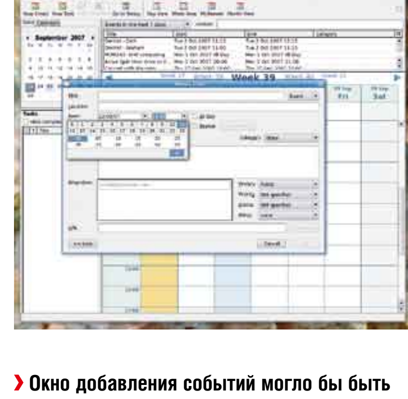
Борьбу возглавила программа *iCal* от Apple, первой включившая поддержку создания нескольких календарей, подписки и онлайн-публикации. Apple также превратила открытый формат *iCal*endar (вопреки названию, это не совсем Apple'овская разработка) в некий стандарт для календарных программ, и подавляющее большинство из них стало загружать и сохранять файлы с расширением *iCal*. А значит, пользователи настольных календарей теперь могут свободно обмениваться файлами формата *iCal*endar друг с другом, и даже публиковать их в сети. Но «истинные» онлайн-календари, среди них и Google'овский, тут же сделали хитрый ход, предложив подписку на *iCal*- или XML-форматированную версию онлайн-календаря для использования на рабочем столе. Единственная проблема – очень уж трудно синхронизировать локальные изменения с онлайн-календарем; многих эта манера *Google Calendar* раздражает с момента его появления. К счастью для пользователей Linux, один из рассматриваемых нами календарей устраняет эту несуразность без труда; другие можно заставить делать это с помощью специальной утилиты. Итак, время выбирать «настольный» вариант календаря!

Mozilla Sunbird

Станет ли *Sunbird* для календарей тем, чем стал *Firefox* для браузеров?

Проект *Sunbird*, начатый как расширение браузера *Mozilla*, стартовал в 2003-м, году радикальных перемен в интернет-пакете: тогда разработчикам пришлось отставить раздувшийся *Mozilla* ради юного выскочки *Firefox*. Пакет-тяжеловес, раздавленный собственной сверхфункциональностью, сменился шустрым окошком, а *Mozilla Calendar Extension* «выпал в осадок». Однако, подобно браузеру *Phoenix*, переродившийся в *Firefox*, *Sunbird* сумел стать отдельным приложением; его философия сходна с *Firefox* – «будем проще, а функциональность расширим благодаря модульной архитектуре». Именно в этом был ключ к успеху *Firefox*. Но web-браузеры – совершенно другой вид приложений. Зачем же календарям модули расширения?

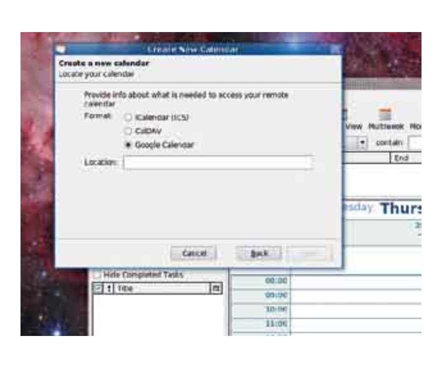




› **Окно добавления событий могло бы быть проще без кнопки More – хороший компромисс между функциональностью и эффективностью.**

Модуль для календаря

Взглянув на короткий список официальных модулей *Sunbird*, не подумаешь, что идея модульной архитектуры здесь развита так же, как в браузере. Например, самый популярный из модулей сворачивает программу в системный лоток. Да. Это все, что он делает. Модуль, выводящий заставку, тоже не вызывает учащенного сердцебиения. Но стоит натолкнуться на нечто под названием *Provider for Google Calendar* (Интеграция с *Google Calendar*), и модульная архитектура тут же вырастает в цене. Для пользователей *Google Calendar* это уже веская причина воспользоваться *Sunbird*. Большинство программ-календарей способны использовать календарь Google для добавления нередактируемых событий в локальный календарь через гиперссылку *iCal*, предоставляемую *Google Calendar*. Но в *Sunbird* этот модуль – двустороннего действия. В отличие от подавляющего большинства других календарей, из *Sunbird* можно добавлять и редактировать события в вашем *Google Calendar*. Поклоннику гугловского календаря этой информации достаточно. Просто установите *Sunbird* и загрузите *Google Calendar Extention*. Инсталляция чуть сложнее, чем для модуля *Firefox*, так как загруженный файл нужно указать вручную, но это недолго. Затем вы называете имя и пароль учетной записи – и тут же получаете прекрасный интерфейс для *Google Calendar*. Единственная проблема (общая для всех календарей) – в *Sunbird* нет способа вручную обновить календарь.



› **Для создания, просмотра и редактирования онлайн-календарей из Sunbird установите расширение *Google Calendar*.**

› **События в *Sunbird* хорошо заметны, только вот места для их текстового описания мало – приходится добывать информацию мышью.**

Не Гуглом единым…

Есть и другие причины для выбора *Sunbird*. Программа «умеет» практически все, что положено программе-календарю. Главное окно делится на панель задач/событий и область календаря. События ярко расцвечены, и хотя интерфейс не претендует на премию в области дизайна, раскладка удобна и понятна. Только вот читать названия событий нелегко, по крайней мере, в помесячном и понедельном вариантах. Для описания каждого события выделяется всего одна строчка, а этого маловато. При попытке чтения обычно приходится дважды щелкать на событии, чтобы развернуть описание, или пользоваться списком предстоящих событий вверху экрана. Было бы неплохо объединить многодневные события (пока событие не выделено, каждый его день представлен как самостоятельный) и показывать по одной строке для каждого в помесячном и понедельном видах. Диалоговое окно добавления событий устроено несложно и работает почти как в *Google Calendar*, где кнопка More (Добавить) добавляет к окну ввода дополнительные поля (например, для описания). А вот поле ввода времени ужасно. Предполагалось сделать его похожим на старый диалог ввода времени Palm OS, с часами вверху окна и минутами – внизу. Но, в отличие от версии Palm, над этим окном придется поломать голову – проще взять и ввести время с клавиатуры. Другая проблема – *Sunbird* не интегрирован в рабочий стол, что, к примеру, затрудняет синхронизацию с PDA. Если вы пользуетесь почтовым клиентом *Thunderbird*, можно инсталлировать расширение Lightning и интегрировать *Sunbird* в единое приложение – отличное решение для тех, кто завязан на этот календарь. А в остальном, главный козырь *Sunbird* – это двусторонний модуль *Google Calendar*. По крайней мере, пока соперники не подтянутся.


LINUX FORMAT Вердикт

Mozilla Sunbird

Версия: 0.5
Сайт: www.mozilla.org/projects/calendar/sunbird/
Лицензия: MPL/GPL/LGPL

» *Непритязательность интерфейса компенсируется простотой и удобством, а также двусторонней связью с Google Calendar.*

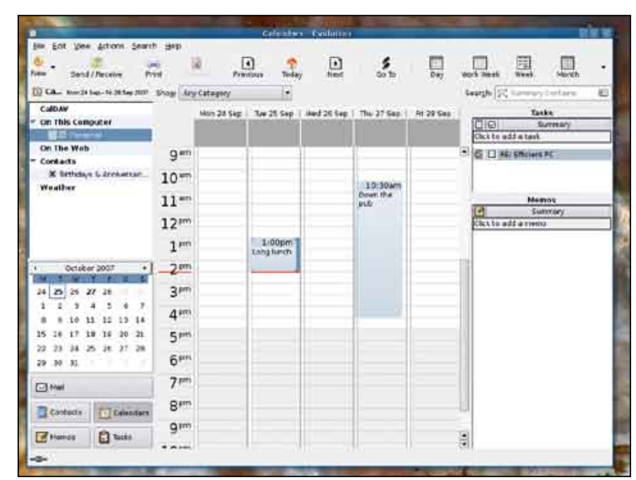

Рейтинг 8/10

Календарь Evolution

Многие ли пользователи *Evolution* о нем знают?

Отделить календарь от PIM-комплекта Gnome невозможно, но это не повод его не рассматривать. Календарь *Evolution* – один из самых близких Linux по духу, и есть прямой смысл держать его рядом со списком контактов и почтовым клиентом. Здесь составляется расписание предстоящих событий – функция, принесшая успех *Microsoft Outlook*. Одна из причин популярности комбинаций почтовых и календарных онлайн-приложений как раз в том и состоит, что из почтового клиента (например, *Evolution*) можно переносить события в календарь, даже не переключаясь на помесячный вид. Кроме того, все эти функции объединяются в единственном приложении. Прекрасный пример интеграции – календарь 'Birthdays and Anniversaries' (Дни рождения и годовщины), который создается по умолчанию и включает важные даты всех людей из вашего списка контактов. Другой пример – если возникла необходимость организовать встречу, то сделать это можно прямо из календаря, а приглашения программа разошлет автоматически с помощью почтового клиента. Единственный недостаток такого подхода в том, что для контактов без электронных адресов приходится создавать фиктивные адреса.

Календарь *Evolution* «понимает» онлайн-контент WebDAV и *iCal* и может синхронизироваться с сервером типа *Exchange* от Microsoft. Как и у большинства рассмотренных календарей, двусторонней связи с *Google Calendar* здесь нет. Зато дисплей гораздо легче читается, чем у других (в особенности *Sunbird* и *KOrganizer*), а само приложение более отзывчиво. С повторяющимися событиями программа управляется отлично (у *Sunbird* с этим проблемы). Нам очень понравилась возможность редактирования событий на месте, прямо там, где они расположены в календарном виде. Просто щелкнул текст в ячейке – и набирай. И выглядят ячейки приятно: мягкие полутона удачно сочетаются с умеренными тенями. За исключением этих скромных дополнений, пользователи *Outlook* при переходе на календарь *Evolution* особой разницы не почувствуют. Единственный недостаток – недостаточно опций для тех, кому нужно нечто большее, чем просто календарь.



› **Календарь *Evolution* один из самых миловидных, а редактирование событий на месте – удобная функция.**



Review name

Версия: 2.11
Сайт: http://gnome.org/projects/evolution/
Лицензия: GPL

» *Отличная интеграция с почтой и списком контактов для пользователей Evolution, но переход требует более веских причин.*

Рейтинг 7/10

Plan

Проще – не значит хуже.

Главная идея *Plan* – возврат к основам. Ликвидированы интеграция с рабочим столом, ресурсоемкие API для Gnome и KDE и груз излишней функциональности, отягощающий другие подобные приложения. Взамен вы получаете простой интерфейс *Motif*, обладающий лишь главными функциями календаря. *Plan* отображает время в 12- или 24-часовом форматах, а форматы дат mmddyy и ddmmyy устроят и бриттов, и янки! Есть понедельный, помесячный и годовой виды, и всех их можно открыть одновременно – такого не умеет больше ни один из рассмотренных нами календарей. Добавить событие просто, а маленькое окошко ввода позволяет ввести несколько событий разом – очень удобно. Радует то, что незачем ломать голову над организацией окна ввода даты: время и число просто вводятся с клавиатуры.

Но простота *Plan* отнюдь не означает, что он бесполезен. Это превосходное решение для тех, кому нужен только календарь. *Plan* имеет и удаленный доступ к календарям, но не ждите поддержки *iCal*: это только способ хостинга главного файла на удаленном сервере, а не средство хитрого преобразования форматов. Зато он позволяет давать доступ к файлам другим пользователям *Plan* и иметь один календарь на группу лиц – предусмотрен даже простейший список допущенных. *Plan* потребляет минимум ресурсов при максимуме функций – эффективнее только командная строка! Здесь есть еще и демон напоминаний, извещающий о предстоящих событиях. Да и технари не забыты: добавить событие в *Plan* можно из командной строки! Например, стоит ввести plan 09261300 -l 1:00 "В пивную" – и вы уже не пропустите эпохальную встречу в любимом пабе. Превосходно!



› **В *Plan* можно открыть несколько разных видов одновременно.**

> «Plan потребляет минимум ресурсов при максимуме функций».



Plan

Версия: 1.9
Сайт: www.bitrot.de
Лицензия: GPL

» *Машина «пожилая», а без полноценного календаря не обойтись? Тогда Plan – для вас.*

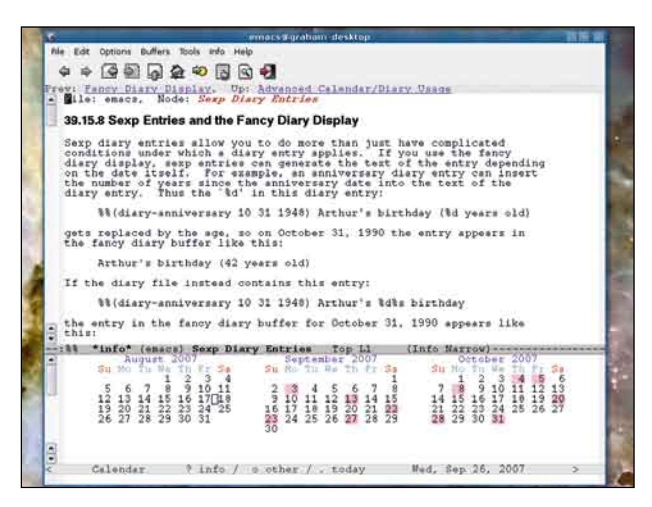
Рейтинг 6/10

Emacs

Нет, это не опечатка. Мы включили *Emacs* в сравнение календарей. Простите, фанаты *vi*!

Все мы знаем и любим *Emacs*. Часто шутят, что этот популярный редактор – скорее операционная система, чем инструмент для работы с текстом. Приложение чудесное: можно играть в игры, управлять файлами, читать почту, компилировать и отлаживать ПО – а заодно уж и редактировать скромные текстовые файлы. Это рабочее место-мечта для программиста. Где-то в нем припрятан и календарь – видимо, чтобы заработавшийся хакер запланировал побольше кофейных пауз. Выберите 'Calendar' в меню Tools (Инструменты), или введите Alt X calendar, если ностальгируете по 70-м. По умолчанию вы получите трехмесячный вид в нижней панели, явно ориентированный на технарей. К примеру, никакой другой календарь не способен на конверсию в систему древних майя; а выводимые сведения включают фазу луны, время восхода и заката, да еще и выходные дни и популярные праздники. В *XEmacs* все это доступно через контекстное меню.

В календарь можно и добавлять события. Создайте простой текстовый файл: в начале каждой строки поместите дату и время, а после них – описание события. Файл сохраните как **~/diary**, и расписание событий готово. Читается такой файл автоматически, а время и описание каждого события появляются в календаре. Сначала это кажется слишком технарским, но освоиться довольно просто – особенно тем, кто подолгу сидит в *Emacs*. Зная *Emacs*, нетрудно смекнуть, что это только цветочки. Здесь в ходу S-выражения (программируемые на Lisp) для настройки условий каждого события – они дают максимальный контроль над повторяющимися событиями, будь то юбилей или день зарплаты, выпавший на выходной: можно вычислить возраст юбиляра или точно узнать, сколько раз событие происходило. На такое способен один *Emacs*, остальные календари и рядом не стоят.



› **Календарь в *Emacs* куда полезнее, чем встроенные тетрисы и пасьянсы.**



Emacs

Версия: 22.1
Сайт: www.gnu.org/software/emacs
Лицензия: GPL

» *Не в меру умный вид? Зато Emacs собрал все, что нужно асоциальному отшельнику-хакеру.*

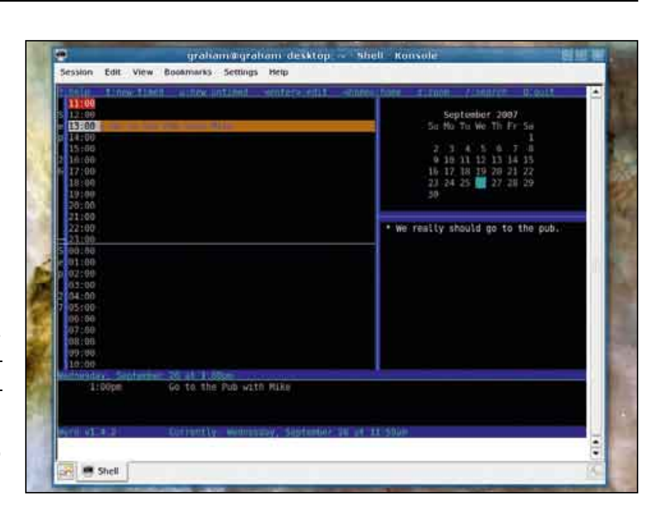
Рейтинг 6/10

Wyrd

Авось шекспиро-пратчеттовское имечко не спугнет вас: штука-то неплохая.

Если *Emacs* вам «не по зубам», но хочется чего-нибудь для командной строки, то, возможно, пригодится *Wyrd*. Это полноценная программа-календарь, работать с которой можно через SSH или даже локально. Управляемый с клавиатуры дисплей не заманит сюда среднего пользователя *Outlook*, но большинство линуксоидов с ходу в нем разберется. Важно, что приложение продолжает развиваться, черпая мощь из *Remind*: популярной мощной программы календаря-будильника, которая задействуется через скриптовые языки и частенько используется для оснащения календарями web-сайтов. Проще всего описать *Wyrd* как оболочку *Remind*, потому что для работы с календарем необходимо усвоить синтаксис *Remind*. Впрочем, чтение man-страницы *Remind* займет всего несколько минут, а гибкость в программировании сложных событий потрясающая.

Для инструмента командной строки, *Wyrd* весьма рационально использует свободное место. Большую часть левой половины экрана занимает прокручиваемая временная таблица. Здесь отображаются события и напоминания, разбитые по дням – что-то вроде вертикальной версии шкалы времени *KOrganizer*. В квадрате справа внизу отображаются события, не связанные со временем, а нижняя часть отведена для описаний. События добавляются нажатием на клавиши: T для события, связанного со временем, или U для не связанного. Запустится ваш текстовый редактор – и в синтаксисе *Remind* можно описать событие. Сохранение изменений перенесет вас обратно в *Wyrd*, и программа отобразит обновленную информацию. Самая крутая функция – Quick Reminders (Мгновенный ввод). Нажмите Q и введите данные, хоть как-то похожие на дату и время, и событие автоматически добавится в календарь. Например, набрав Рвануть с Майком в пивняк завтра в 1 час дня, вы получите в календаре событие, назначенное на 13:00 следующего дня. Ну, чем не продвинутое приложение? Даром что в командной строке.



› **Благодаря *Curses*, пользовательский интерфейс этого инструмента командной строки действительно весьма гибок.**



Wyrd

Версия: 1.4.2
Сайт: www.eecs.umich.edu/~pelzlpj/wyrd/
Лицензия: GPL

» *Замечателен для соединений с малой скоростью, но сложность изучения и ограничения на взаимодействие вряд ли завоюют толпы почитателей.*

Рейтинг 5/10

KOrganizer

Наверное, 'Kalendar' уже занято…

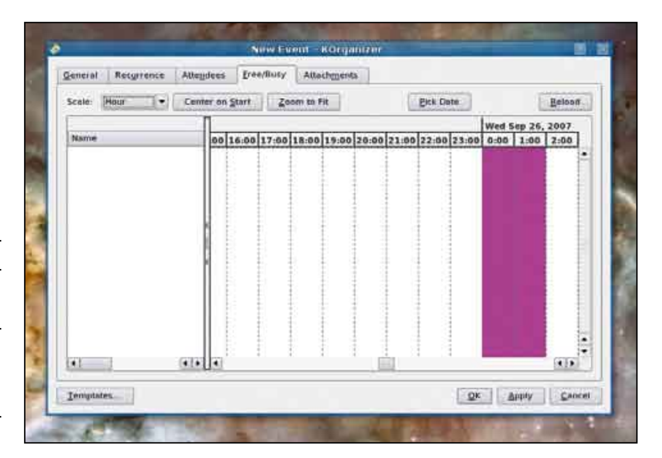
Все, что делает Gnome, должен делать и KDE. *KOrganizer* – ответ KDE на календарную приставку к *Evolution*. По умолчанию календарь встроен в PIM-комплект *Kontact* и обладает ровно теми же преимуществами интеграции, что и его двойник из Gnome. Можно разделять список контактов и добавлять события, не выходя из программы. Но благодаря функциональности *KPart* календарь запускается и без тяжких кандалов PIM или необходимости ввода почтовых адресов: программа та же, только ее интерфейс вынимается из пакета и помещается в стандартное окно KDE. Всего и надо, что запустить *KOrganizer* отдельно. Это солидное преимущество для тех, кто не любит неповоротливые пакеты вроде *Evolution* или *Kontact*, которое существенно снижает ментальную и физическую нагрузку при работе с ПО. Можно даже щелкнуть на файле календаря в *Konqueror*, и календарное приложение KDE запустится в окне браузера.

KOrganizer напоминает *iCal* больше других приложений, участвующих в нашем сравнении. Например, в нижней левой панели окна можно добавлять и изменять календари, в точности как в календаре от Apple. Однако функциональность *KOrganizer* существенно шире. Здесь можно загружать календари с *Exchange*-сервера Microsoft, его свободной альтернативы *OpenXchange*, *eGroupware*. Возможны варианты *Groupwise* от Novell, а также обычные XML и *iCal*, локальные и удаленные (хотя это будет посложнее). *Evolution* хоть и располагает сходными функциями, но пользователю их не показывает: широкий выбор, кроме преимуществ, обладает и недостатками. В *KOrganizer* каждый календарь снабжен диалоговым окном ресурсов, где от изобилия параметров рябит в глазах. Здесь можно настроить частоту обновления, адреса выгрузки и загрузки, установить интервал сохранения. Мечта продвинутого пользователя, но для того, кто ищет простое приложение, в *KOrganizer* много лишнего.

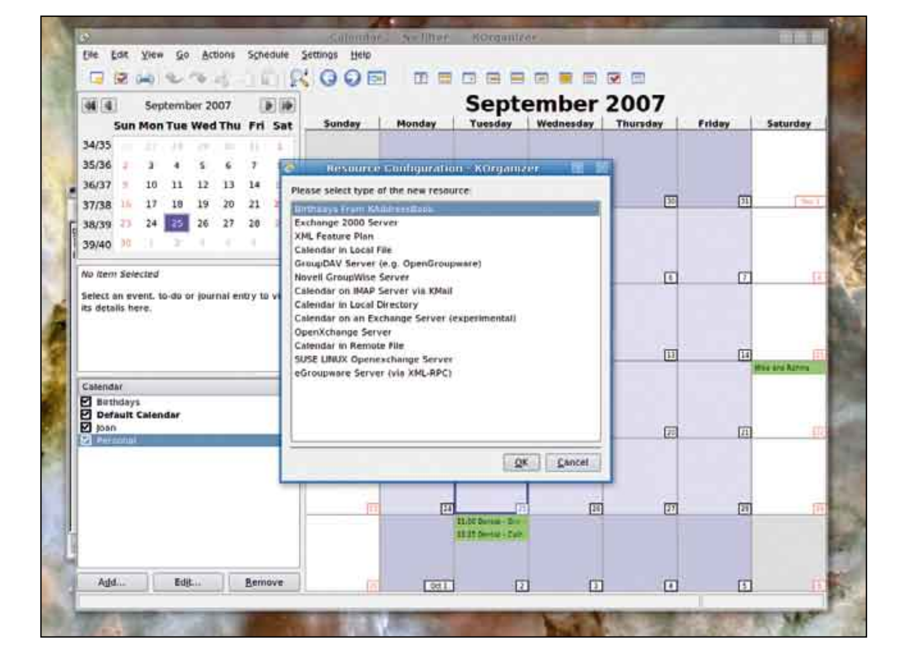
Рисуем шкалу времени

После создания или загрузки календарей события отображаются в простом, разделенном на блоки, прямоугольнике в окне программы. Несмотря на то, что календарь – часть KDE, внешний лоск здесь напрочь отсутствует, все просто и по-деловому. Редактировать события на месте нельзя, но можно, по крайней мере, сдвигать границы событий для изменения времени (как и в *Evolution*, и в *Sunbird*). Есть обычный набор видов, но наши взгляды зацепила Шкала времени. Каждый календарь отображается слева, даты выстроены вдоль оси 'Y'. События занимают блоки по горизонтали, поэтому «накладки» заметны сразу. Прекрасная идея; правда, хотелось бы добавить масштабирование вида (по умолчанию, блоки слишком малы), к тому же «перескакивать» можно только на целую неделю. Не помешала бы возможность передвигать панель дат, а также отодвигать вид, чтобы разглядеть картину событий. С этими поправками Шкала времени могла бы стать подлинно инновационной функцией. Но есть у нее и другое применение. На панели New Event (Новое событие) Шкала времени используется для того, чтобы точно указать время, когда вы заняты, и выбор функций здесь существенно шире, чем в главном окне. Панель новых событий оформлена в духе KDE, здесь можно назначить частоту повторений, составить список приглашенных и привесить необходимые вложения (вот это нам понравилось). Но для напоминания о визите к зубному такое великолепие явно излишне.

KOrganizer мог бы стать лучшим выбором для пользователей KDE, особенно тех, кто пользуется *KMail* и *KAddressbook*. Но им, похоже, пренебрегают. Богатый выбор функций – это прекрасно, а вот их перебор отгоняет желающих. Несомненно, программа мощная, особенно для офисных или корпоративных пользователей: с функционалом *KOrganizer* большинству календарей в нашем Сравнении не тягаться. Надеемся, что KDE 4 исправит недостатки интерфейса.



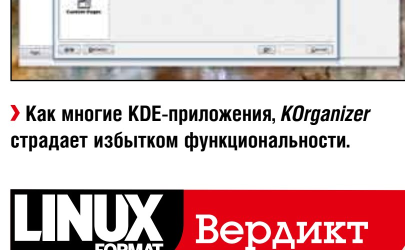
› **Шкала времени используется для добавления событий и просмотра их общей картины в главном окне программы.**



› ***KOrganizer* – лучший выбор, когда нужна взаимосвязь с корпоративным сервером.**



› **Как многие KDE-приложения, *KOrganizer* страдает избытком функциональности.**



LINUX FORMAT Вердикт

KOrganizer

Версия: 3.5.6
Сайт: http://kontact.kde.org/korganizer
Лицензия: GPL

» *У программы богатейший набор функций, достойный лидера. А вот про интерфейс этого не скажешь.*

Рейтинг 7/10

Календари

Вердикт

Mozilla Sunbird 8/10

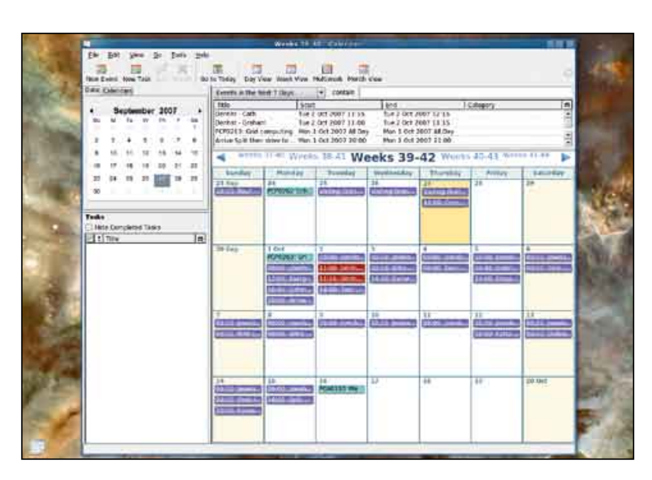
Победитель – *Sunbird*, и не только потому, что это единственная программа, имеющая двустороннюю связь с *Google Calendar* (хотя и через отдельный модуль). *Sunbird* нашел верный баланс между простотой и функциональностью. Изучать программу почти не требуется, события можно добавлять тут же, с ходу. Отсюда и лавры.

Тем не менее, *Sunbird* – не идеал. Хотелось бы видеть суперприложение, вобравшее лучшие качества каждой из программ-участниц нашего **Сравнения** – всем календарям календарь. Например, у *Evolution* фантастическое «редактирование на месте». Единожды попробовав, вы ловите себя на том, что щелкаете по текстовым сообщениям в других программах в расчете на этот эффект – тщетно, чуда не происходит. Внешность *Evolution* также вне конкуренции, рядом с ней остальные приложения выглядят примитивными.

От *Plan* и *Wyrd* пригодилось бы управление событиями из командной строки. Это не пижонство, а весьма удобный способ добавления событий прямо из консоли, заодно открывающий возможность создания событий с помощью скриптов. Аналогично, из почтенного *Emacs* мы бы взяли алгоритмическое создание событий. Пусть этой функцией трудно воспользоваться для чего-нибудь, кроме составления сложных корпоративных графиков, но столь крутая возможность обязана существовать, а уж пользователи найдут ей применение – хотя бы для создания событий типа «сборище в пивной».

Наконец, от *KOrganizer* хотелось бы взять Шкалу времени и способность добавлять вложения к предстоящим событиям. «В одной посуде» все это завоевало бы 10/10, и у нас был бы календарь-совершенство. Тот факт, что победитель заработал лишь 8/10, свидетельствует, что программы, участвовавшие в **Сравнении**, далеко не исчерпали резервы роста. Думаем, *Sunbird* удержит первенство. Надеемся также, что Gnome и KDE ответят на вызов. LXF

> «Уникальные свойства всех календарей “в одной посуде” завоевали бы 10/10.»



› Sunbird предлагает пользователю комфорт и самые полезные функции, не осложняя каждодневного использования.

Ваше мнение

Считаете ли вы, что мы слишком зациклились на Google Calendar? А может, мы упустили какой-нибудь ценный календарь? Сообщите нам! Главные «календарные» проекты всегда на виду, но ведь в мире Open Source всегда может оказаться, что кто-то разработал свой собственный. Если это удалось именно вам, будем рады узнать об этом. Не теряйтесь, пишите на letters@linuxformat.ru.

Calendar от Google – всем

Может быть, *Sunbird* – единственная программа, оснащенная встроенной двусторонней связью с *Google Calendar*, но существует решение, доступное любой другой программе с поддержкой формата *iCalendar*. Свободный инструмент под названием *GCALdaemon* – это небольшой Java-сервер, работающий в фоновом режиме между календарным приложением и сервером Google.

Заменить адрес сервера Google адресом машины, на которой запущен *GCALdaemon* – вот все, что нужно для настройки клиента, да и сервер в настройке несложен, с единственным файлом конфигурации и четким механизмом «вопрос/ответ». Мы поставили его на той же машине, где проводилось тестирование, и не имели никаких проблем с чтением/записью в нашем онлайн-календаре, как из *Evolution*, так и из *KOrganizer*. Единственная возможная проблема, с которой вы можете столкнуться, если подключение к Интернету не постоянное – придется составить небольшой скрипт для подключения/отключения от сервера Google. Естественно, для работы *GCALdaemon* понадобится адрес Gmail и Java 1.5 (или выше) Runtime Environment. http://gcaldaemon.sourceforge.net.

Linux Format 97



Пользование программой-календарем позволит вам организовать всевозможные проволочки!

Наш «лучший друг новичков» Энди Ченнел написал углубленное введение в использование *Sunbird*, *Lightning* и *Provider* в LXF97.

Таблица функций

	Чтение из Google Calendar	Запись в Google Calendar	Поддержка iCal	WebDav	Дневной вид	Понедельный вид	Помесячный вид	Годовой вид	Синхр.с PDA	Интеграция с Exchange	Журнал	Поиск	Поддержка нескольких календарей
Emacs	☒	☒	☒	☒	☒	☒	✓	☒	☒	☒	✓	✓	✓
Evolution	✓	☒	✓	✓	✓	✓	✓	✓	✓	✓	☒	✓	✓
KOrganizer	✓	☒	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Sunbird	✓	✓	✓	✓	✓	✓	✓	✓	☒	☒	☒	✓	✓
Plan	☒	☒	☒	☒	✓	☒	✓	✓	☒	☒	☒	✓	☒
Wyrd	☒	☒	☒	☒	✓	☒	✓	☒	☒	☒	☒	✓	☒



Ежемесячная сводка новостей дистрибутивов Linux



ЛАДИСЛАВ БОДНАР
основатель, редактор, начальник и сотрудник DistroWatch.com.

Зрелый Linux

Неприятная штука – я должен сделать признание: я не знаю, как собрать ядро Linux. Или, точнее, я больше не знаю, как его собрать.

Помню ранние дни моего исследования нашей любимой ОС. Ознакомившись с основами, я, недолго думая, решил попробовать на зуб нечто более захватывающее, вроде замены рабочего ядра Linux моего дистрибутива собранным собственноручно. Затем я, затаив дыхание, смотрел, как прекрасно загружается система, несмотря на то, что ей подсунули ядро, собранное преданным, но абсолютным дилетантом.

Небольшое и быстрое

Я давно уже не дилетант. За прошедшие годы бывали периоды, когда я пересобирал ядро практически ежедневно – иногда из-за добавления нового устройства требовалось включить дополнительную функциональность, а в других случаях просто для забавы или обучения. В те дни у меня было ядро, которым я по праву гордился – небольшое, быстрое и собранное именно для моей машины. Я знаю свой путь в древе menuconfig как немногие другие.

Но теперь я уже не забочусь об оптимизации своего ядра. Возможно, это знамение времени: тогда, на стыке веков, большинство пользователей Linux были энтузиастами, Linux – скорее игрушкой, а работа с ним – хобби. Но с ростом числа обычных пользователей, устанавливающих Linux на свои машины, и при изобилии простых в использовании дистрибутивов, Linux уже больше не безделица: он стал инструментом, операционной системой для повседневного использования. Linux созрел.

ladislav.bodnar@futurenet.co.uk

Власть над миром в стиле Ubuntu

Заявив выход Hardy Heron в апреле 2008 и его длительную поддержку, Gutsy несет новые функции рабочего стола.

Зримый успех Ubuntu как настольного дистрибутива Linux озадачил многих в индустрии. Что же такое есть у этого проекта, что вызывает настолько позитивные ответные вибрации в Интернете? И как случилось, что кое для кого Ubuntu стал синонимом Linux?

Некоторые Linux-компании и команды разработчиков утверждают, что единственная причина растущей популярности Ubuntu – доступность CD, поставляемого в любую точку мира бесплатно. Кабы не бездонный кошель Марка Шаттлворта, говорят они, Ubuntu никогда бы не видать подобного взлета. Но большинство пользователей Linux, вероятно, согласится с тем, что проект, прибегающий к таким аргументам, просто пытается замазать собственные недостатки. Хоть и верно, что инициатива ShipIt увеличила число пользователей Ubuntu, но вряд ли одних лишь красочных CD, содержащих некие абстрактные африканские слова о человечности, было бы достаточно, чтобы соблазнить тысячи пользователей сменить свою старую ОС на Ubuntu.



› **Ubuntu 7.10 – теперь еще легче в использовании с надежной утилитой настройки X.**

Главнейшие факторы, которые создали феномен Ubuntu в Linux – это, на самом деле, удобство и предсказуемость. Превосходный помощник миграции данных и настроек и доступность проприетарных драйверов для множества видео- и сетевых карт гарантируют большинству пользователей приятное первое впечатление от дистрибутива. Последний релиз Ubuntu 7.10 «Gutsy Gibbon» [Дерзкий гиббон] превзошел всех остальных в предоставлении безотказной настройки видеокарты и монитора, благодаря новой концепции, названной *BulletProofX*.

Что до предсказуемости, то многие дистрибутивы часто избегают ее, как присущую миру проприетарных программ. Но для множества организаций предсказуемость жизненно важна. Когда выйдет следующая версия? Долго ли она будет поддерживаться? Последнее объявление Ubuntu о выходе следующей версии «Hardy Heron» [Храбрая цапля] в апреле 2008 описывает его как релиз с длительной поддержкой [Long Term Support (LTS)]: его настольная редакция будет получать обновления безопасности до апреля 2011, а серверная – до 2013. Столь ясная политика отлично работает на завоевание доверия потенциальных пользователей, планирующих выбор операционной системы.

Естественно, Ubuntu не идеален. Но, как отметила в своем интервью Эухения Лоли-Квири [Eugenia Loli-Query] из OSNews.com, он сейчас предлагает лучшее качество «из коробки» среди всех дистрибутивов Linux. Твердое стремление лидеров и разработчиков Ubuntu принести Linux миллионам пользователей уже изменило компьютерную арену. Если так пойдет и далее, не удивляйтесь, если в скором будущем Ubuntu станет доминирующей ОС, а не только доминирующим дистрибутивом Linux.

www.ubuntu.com

Просто лучший?

Mandriva Linux 2008 – новые модули DrakConf, CompFusion, пре-релиз KDE 4.

Mandriva Linux 2008 – весьма ожидаемый релиз. Это не просто первое серьезное обновление дистрибутива за год; версия 2008 также устанавливает планку для других основных дистрибутивов, выпускающих релизы вслед за ней, поскольку содержит новое ядро 2.6.22, Gnome 2.20 и предварительную версию KDE 4, для заинтересованных бета-тестеров.

В списке нововведений Mandriva Linux 2008 можно найти долгожданную функцию записи на NTFS (включена по умолчанию) и поддержку из коробки большого числа беспроводных сетевых карт, включая некоторые проблемные USB-экземпляры. Безопасность системы усилена включением AppArmor от Novell, программной оболочки безопасности, подобной SELinux от Red Hat. По части трехмерного рабочего стола, *Compiz* и *Beryl* заменены недавно объединенным *CompFusion*, а в качестве альтернативы доступен Metisse. Также на установочном DVD находятся проприетарные драйвера ATI и Nvidia.

После давнишнего снижения популярности по сравнению с Mandrake, Mandriva усиленно старается вернуть былое после нескольких лет переменных успехов – и как компания, и как дистрибутив. Если судить по релиз-кандидатам версии 2008, мы увидим прекрасный, качественный настольный дистрибутив, вполне способный соперничать и с Ubuntu, и с OpenSUSE в смысле дружелюбности и инноваций.

www.mandriva.com



› Тестовая сборка Mandriva Linux 2008 выглядит хорошо: обновлены оформление и тема рабочего стола.

Товарищ Linux

ALT Linux 4.0 – новый дистрибутив на кириллице… Из России с ALT.

Дистрибутив ALT Linux, разработанный преимущественно для русскоязычного рынка, стартовал в 2001 г. как небольшой проект сообщества. Сперва он концентрировался на улучшении поддержки русского языка в Mandrake Linux, но разработчики, накопив опыт, начали собирать собственные пакеты и интегрировать их в различные настольные и серверные издания своего дистрибутива. Теперь проект стал коммерческой компанией, предоставляющей поддержку, обслуживание и помощь в миграции, и продолжает периодически выпускать новые версии своего флагманского продукта. ALT Linux 4.0 Personal Desktop вышел в конце августа.

Образ DVD весом 4.4 ГБ доступен для бесплатной загрузки с FTP-сервера дистрибутива [а также в LXF97, – *прим. ред.*]. Он может загружаться в Live-режиме, если кто-то хочет поэкспериментировать с операционной системой без последствий, или устанавливаться на жесткий диск при помощи простого графического установщика. По умолчанию ALT Linux настраивает стандартный рабочий стол KDE. Также он может похвастаться превосходной автоматической настройкой устройств, поскольку предоставляет популярные проприетарные графические драйвера и драйвера для большинства беспроводных карт.



› Рабочий стол и его утилиты доступны также на английском.

Уникальной функцией ALT Linux 4.0 является *ALTerator*, графическая утилита настройки, доступная в качестве отдельного приложения или как модуль с web-интерфейсом, доступный по безопасному соединению. Она позволяет легко настроить различные параметры служб и устройств, а также предоставляет способ обновления системы свежими пакетами и исправлениями безопасности. Также прилагается *Synaptic* – графическая система управления пакетами, адаптированная для работы с RPM.

Сообщества форумов

В LXF90 мы подсчитали число зарегистрированных пользователей на форумах популярных дистрибутивов; теперь мы делаем это вновь. В абсолютном измерении таблицу возглавил бы Ubuntu; но форум PCLinuxOS установил рекорд по приросту членов – 144%.

Ранг	Дистрибутив	Янв. 2007	Сент. 2007	Изменение	Форум
1	PCLinuxOS	4,800	11,700	+144%	pclinuxos.com/forum
2	Freespire	10,100	18,600	+84%	forum.freespire.org
3	PC-BSD	5,700	10,300	+81%	forums.pcbsd.org
4	Ubuntu	215,200	376,200	+75%	ubuntuforums.org
5	Arch Linux	7,500	10,600	+41%	bbs.archlinux.org
6	Kubuntu	14,000	21,000	+33%	kubuntuforums.net
7	OpenSUSE	41,800	52,200	+25%	suseforums.net, forums.suselinuxsupport.de
8	Fedora	84,500	100,000	+18%	forums.fedoraforum.org
9	Xandros	24,200	28,000	+16%	forums.xandros.com
10	Mandriva	54,800	58,800	+7%	forum.mandrivaclub.com, mandrivausers.org
11	Gentoo	103,000	109,000	+6%	forums.gentoo.org
--	Sabayon	Данных нет	8,500		sabayonlinux.org/forum
--	Puppy	Данных нет	7,000		murga-linux.com/puppy
--	Mint	Данных нет	3,400		linuxmint.com/forum
--	Zenwalk	Данных нет	3,400		support.zenwalk.org

Годовой хит-парад

10 самых посещаемых страниц на DistroWatch.com с октября 2006 по сентябрь 2007 (среднее число визитов в день).

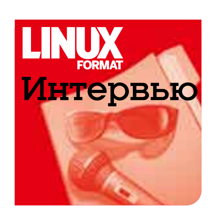
	Дистрибутив	Число визитов	
1	Ubuntu	2,572	↓
2	PCLinuxOS	1,901	↑
3	OpenSUSE	1,810	↓
4	Fedora	1,410	↓
5	MEPIS	1,010	↓
6	Debian	983	↓
7	Sabayon	976	↓
8	Mandriva	882	↓
9	Mint	770	↓
10	DamnSmall Linux	712	↓

» **DistroWatch.com** следит за популярностью дистрибутивов, основываясь на числе визитов на страницу каждого дистрибутива. Хотя оно не совпадает с действительным числом установок, но показывает, какие дистрибутивы более популярны за определенный промежуток времени. LXF

Персональный ПОДХОД



Как вы думаете, можно создать собственный дистрибутив онлайн, всего-навсего загрузив ISO с нужными для этого пакетами? С помощью *Revisor* – можно! Говорит Джек Эбаутбул из Fedora.



В тот день, когда этот выпуск *Linux Format* появится в продаже, Fedora 8 уже, вероятно, будет доступна для скачивания. Но за пару месяцев до ее выхода, запланированного на 8 ноября 2007 г., Ник Вейч встретился с «Крестным отцом» посланников Fedora, Джеком Эбаутбулом [Jack Aboutboul], чтобы побеседовать о грядущем релизе и о будущем Fedora.

Джек начинал как пользователь Linux лет десять назад, был разработчиком на начальной стадии Gentoo Linux и членом-соучредителем LUG Университета Сиракуз. Кроме того, в качестве первого разработчика EduGate, пакета управления ресурсами учебного заведения, недавно, в конце 2007 года, он основал Фонд Джека Эбаутбула [Jack Aboutboul Foundation], чья миссия заключается в «создании мостов между открытой технологией, открытым содержанием, искусством и образованием в целенаправленной попытке обогатить жизнь взрослых и детей во всем мире; а также поддержать творческое начало, присутствующее в каждом человеке». http://jackfoundation.com.

Linux Format: Расскажите, почему вас зовут «Крестным отцом»? Вас застали за прибиванием лошадиной головы к чьей-то кровати [намек на эпизод романа Марио Пьюзо «Крестный отец», — *прим. пер.*]?

Джек Эбаутбул: За этот пышный титул следует благодарить Грега ДеКенигсберга [Greg DeKoenigsberg], директора по связям с сообществом Red Hat. Моя официальная должность – инженер сообщества [Community Engineer] в Red Hat, и большинство моих усилий сосредоточено на Fedora Project. В прошлом я содействовал созданию некоторых инициатив сообщества Fedora, например, Fedora BugZappers Triage Team; я создал и помогал в работе FUDCon [Fedora Users and Developer Conference, Конференция Пользователей и Разработчиков Fedora] http://fedoraproject.org/wiki/FUDCon. И продолжаю заниматься тем, чем всегда занимался – работой над интересными проектами.



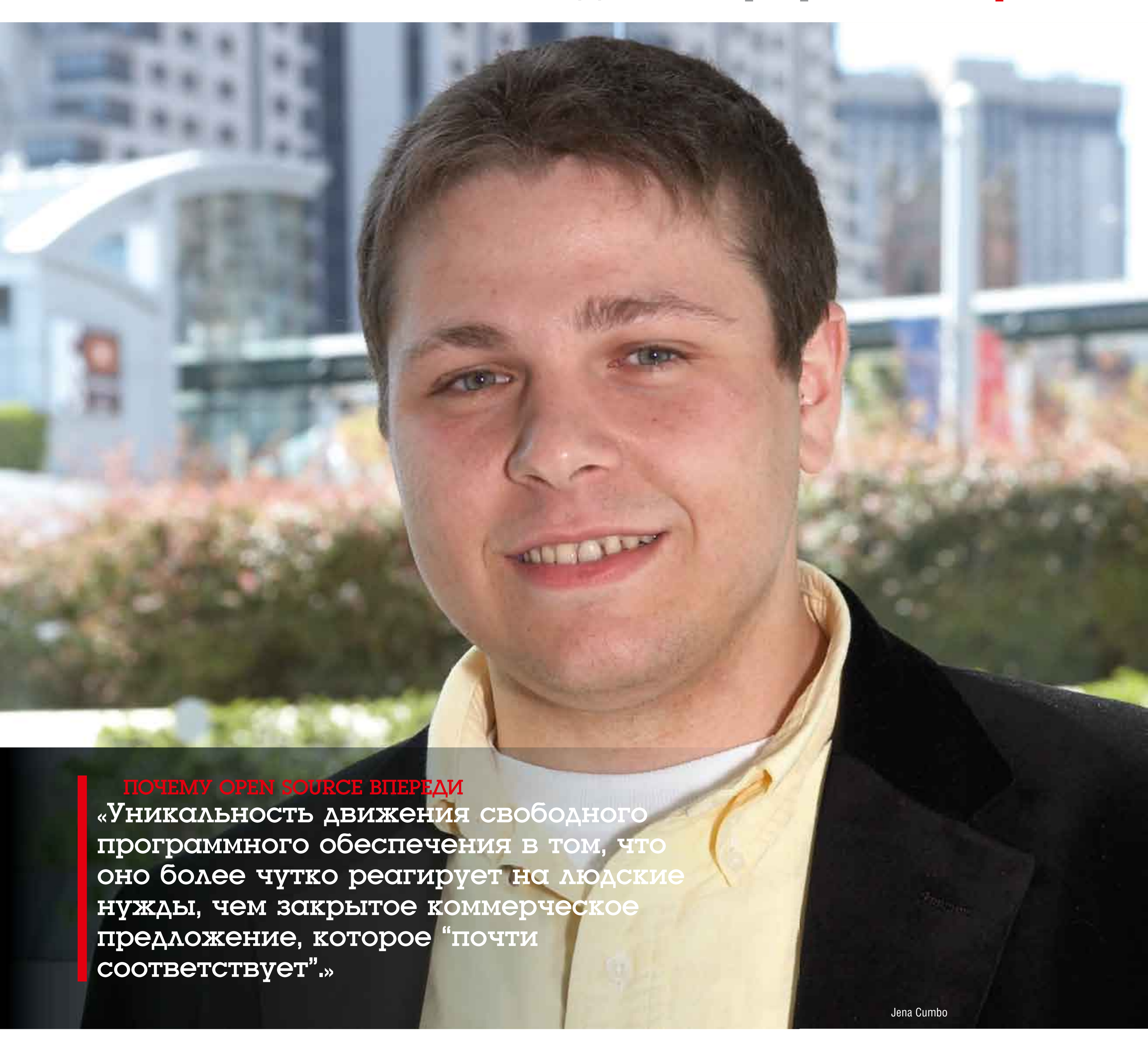
LXF: Над интересными проектами какого рода?

ДжЭ: Этим летом мы работали над *Revisor*, http://revisor.fedoraunity.org, графическим инструментом перестройки; мы перерабатывали его в web-сервис, чтобы можно было войти в него, где бы вы ни находились, создать свои собственные настройки, и чтобы он создал их для вас и отправил вам сообщение по электронной почте, когда ISO будет готов к загрузке. Надо отдать должное всем участникам этого проекта за их тяжкий труд. *Revisor* – это действительно классно, потому что больше всего пользователи жалуются на массивность дистрибутивов, им не хочется загружать все то, что включено в дистрибутив, если им нужно всего несколько пакетов. Некоторые на самом деле хотят, чтобы он работал на маршрутизаторе или вроде этого; с помощью *Revisor* вы можете выйти в режим онлайн и собрать, например, ISO размером 50 МБ, и он пропингует и сообщит, что все готово к загрузке. Его можно применять с разными целями, не только тем, кто занимается бизнесом, так что инструмент классный. Мы работали над ним изо всех сил и на прошлой неделе выложили бета-версию. (Подробнее о *Revisor* вы можете узнать в спецрепортаже этого месяца).

LXF: Для желающих заниматься индивидуальным проектом это очень полезно, но захотят ли компании, использующие Red Hat, его тоже использовать?

ДжЭ: Конечно! Основной целью была возможность предложить поддержку для всех этих самопальных дистрибутивов. Red Hat привержена к новой парадигме программ в мире бизнеса. У нас в вестибюле огромными буквами написано «Стать компанией, определяющей технологию XXI века». Впервые это увидев, я подумал: «ну дают!», однако время идет, и я нахожу эту цель все менее и менее смешной – это весьма серьезная заявка.

Это ведь очень хорошо, когда можно создать такую программу, позволяющую легко настроить дистрибутив под требования каждого отдельного пользователя. С помощью инструментов, которые мы предлагаем, становится намного проще самим создать то, что вам нужно, и начать этим пользоваться; нас подвигнул тот факт, что пользователь всегда лучше нас знает, что именно ему надо – это очевидно. Что касается использования в бизнесе того, чем мы занимаемся, мы хотим достичь такого момента, когда мы сможем создавать стеки для хостинга. Мы работаем с разными компаниями, предоставляющими web-сервисы над этим проектом, поэтому мы хотим, чтобы он был в состоянии выдать для вас образ – вы создаете подборку по Fedora с какими-то своими дополнительными пакетами и, хочется надеяться, планируете со временем включить их в Fedora. Знаете, ведь идеал не всегда достижим, и неплохо иметь возможность создать нечто свое, поместить в виде образа ISO на некое отведенное место и запускать его оттуда. Захочет ли



ПОЧЕМУ OPEN SOURCE ВПЕРЕДИ

«Уникальность движения свободного программного обеспечения в том, что оно более чутко реагирует на людские нужды, чем закрытое коммерческое предложение, которое “почти соответствует”.»

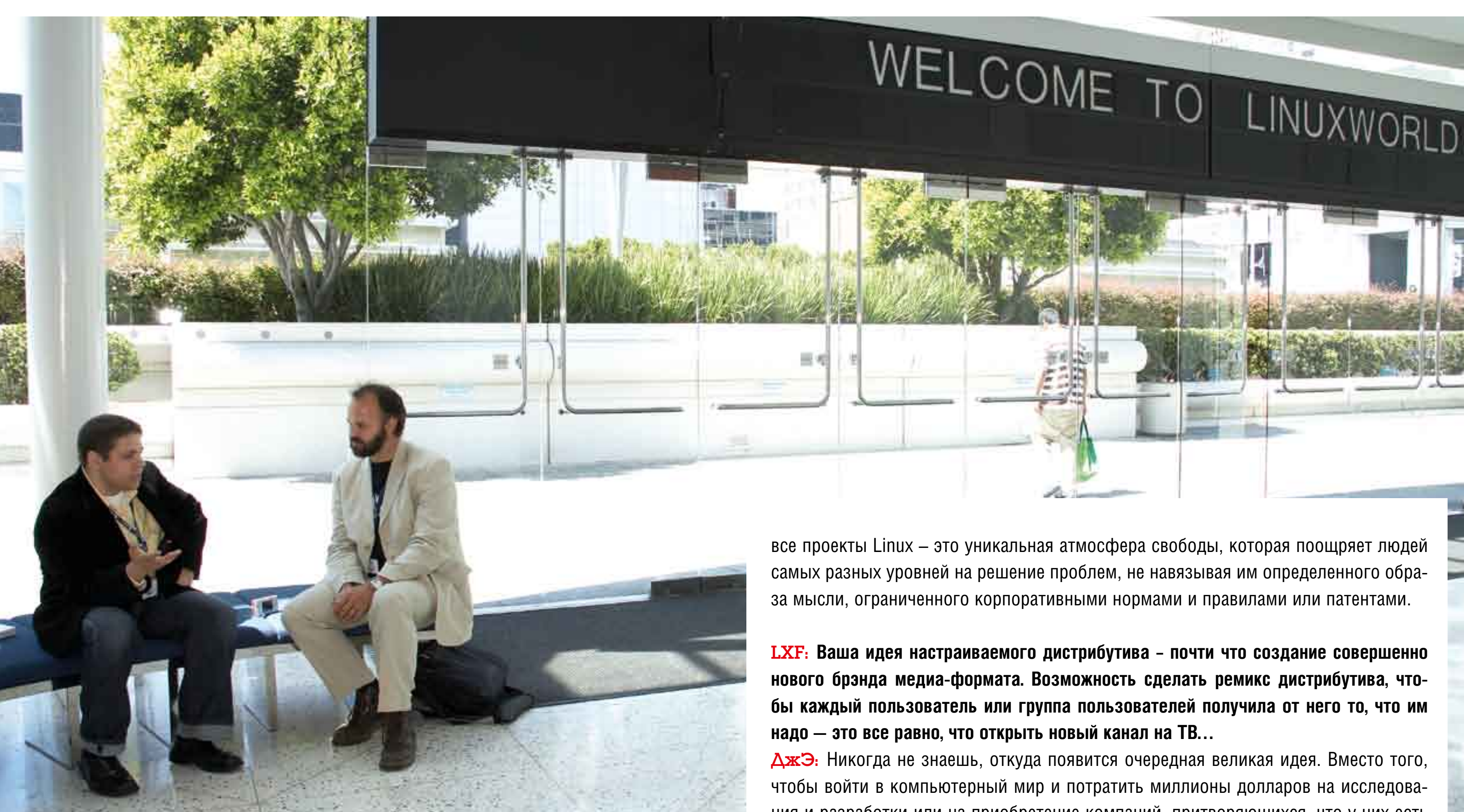
Jena Cumbo

Red Hat развивать это все дальше, чтобы предоставлять услуги по поддержке, или будет продавать что-то другое – от меня не зависит, но все же это стало возможным, такая возможность есть. И я не думаю, что мы упустим шанс воспользоваться этой возможностью.

LXF: По-моему, Red Hat и Fedora раньше считались одними из самых сложных в настойке дистрибутивов. И я думаю, именно по этой причине некоторые обращались к другим дистрибутивам, однако сейчас, благодаря разработке, которой вы занимаетесь, эта ситуация изменится?

ДжЭ: Ну, по правде говоря, любые настройки сложно делать! Я пытался поиграть с разными дистрибутивами, чтобы самому посмотреть, насколько сложно изменить тип их настройки, и увидеть, что вообще можно сделать. Вот Клаус Кноппер [Klaus Knopper] просто вызывает благоговение: я не представляю, как он это делает, это просто громадная работа – заставить функционировать такое. Knoppix – превосходный дистрибутив, и в основном это работа одного человека; то, что он сумел все это собрать, это уже подвиг. А затем, пару месяцев спустя, все уже вовсю пытаются сравняться с ним, и стандартизировать весь этот процесс для нас, и частью проблемы стало наличие двух конфликтующих наборов инструментов для создания «живых» образов, поэтому мы начали со штуки под названием Pilgrim. А потом мы стали писать нечто принципиально новое, именуемое в то время Live CD Tools, и столкнулись с такой дилеммой: если мы выберем одно, то взбесятся одни люди, а если выберем другое, то взбесятся другие люди, и что же нам делать? Фактически же ни одно из этих приложений не соответствовало в полной мере тому, что мы собирались делать. Вот так и появился *Revisor*. Просто люди из сообщества сказали: «Эй, мы пишем новый инструмент». На FUDcon мы кого пригласили, кому заплатили, и подумали: «Почему бы не использовать *Revisor* в качестве своего проекта в выходные?» Именно тогда мы перешли от того, чего еще не существовало, к тому, что уже наполовину работало; и, очевидно, с того момента оно и начало улучшаться. Это – блестящий пример того, насколько эффективным может стать обращение к сообществу за содействием.

Раньше у нас была та же проблема – в прошлом году, когда я встречался с LXF, мы беседовали о намерении сделать Fedora полностью открытой. Это была основная трудность: кое-кто в Red Hat считал, что это плохая идея, и что мы просто выпустим на волю всех чертей из ада и потеряем контроль над тем, что обязаны »



» контролировать. Были разработчики Fedora, напуганные мерой ответственности, которую надо было принимать на себя. Надо было делать огромную работу. Надо было позаботиться о проблеме интеллектуальной собственности (ИС) – на этом мы чуть хребет себе не проломили.

LXF: Помню, было множество пакетов Fedora с некорректной или устаревшей лицензионной информацией. Продраться через все эти пакеты для проверки данной информации было просто подвигом Геракла...

ДжЭ: И все это делалось вручную! Этим мы тоже начали заниматься на FUDcon. Мы установили в лаборатории 20 машин, народ уселся и принялся за просмотр лицензий, проверку пакетов, чтобы убедиться, что все соответствует стандартам, выбранным нами для пакетов. Это был интенсивный ручной труд, но мы сделали это! Думаю, вряд ли кто-то может осознать масштабность поставленной задачи, пока сам не примется за какой-нибудь крупный проект – например, дистрибутив. В тот день, когда мы это сделали, мы объявили об этом во всеуслышание, и многим просто не верилось в наше достижение!

Благодарнее всех оказались ребята из других софтверных компаний, которые занимались созданием решений на нашей базе – мы значительно облегчили их жизнь. Выполнение этой задачи укрепило чувство общности между нами и теми, кто полагался на нас. Creative Commons стали основным фактором, который мы до того момента не очень-то брали в расчет, но он действительно помог нам!

LXF: Я говорил с Йоном Филипсом [Jon Phillips] о том, как *Linux Format* мог бы помочь сообществу Linux в целом с документацией. У нас огромный каталог статей и изображений, и мы ломаем голову над тем, как бы издать все это без ущерба нашему тиражу...

ДжЭ: Выгоды в конечном итоге будут безграничны – если учебники LXF включать в дистрибутивы, это разнесет ваши слова повсюду! По-моему, начиная с самых ранних реализаций дистрибутива Fedora, мы всегда хотели позиционировать себя как платформу для инноваций. Это важно вне зависимости от того, выдвигаем ли мы инновацию сами, или просто помогаем кому-то с его инновацией. Мы даем людям в руки инструменты, вот чем мы занимаемся.

Вероятно, есть два основных источника инновации. Очевидно, есть технические инновации; здесь мы – лидеры, мы более всего известны благодаря нашему техническому превосходству. И мы намерены так держать: ни за какие коврижки мы не откажемся от своей основной компетенции.

А еще есть атмосфера познания, которую поощряет не только Red Hat, но и все проекты Linux – это уникальная атмосфера свободы, которая поощряет людей самых разных уровней на решение проблем, не навязывая им определенного образа мысли, ограниченного корпоративными нормами и правилами или патентами.

LXF: Ваша идея настраиваемого дистрибутива – почти что создание совершенно нового брэнда медиа-формата. Возможность сделать ремикс дистрибутива, чтобы каждый пользователь или группа пользователей получила от него то, что им надо — это все равно, что открыть новый канал на ТВ...

ДжЭ: Никогда не знаешь, откуда появится очередная великая идея. Вместо того, чтобы войти в компьютерный мир и потратить миллионы долларов на исследования и разработки или на приобретение компаний, притворяющихся, что у них есть какие-то сумасшедшие проекты по ИС или по программам, которые никогда не дадут результата, пригодного для включения в бизнес-модель. Зачем это надо? Проще подтолкнуть сообщество. Мы можем откинуться в кресле и расслабиться, а великие идеи проявятся благодаря процессу обмена идеями внутри сообщества.

Майкл Тиманн [Michael Tiemann], вице-президент по взаимодействию с Open Source в Red Hat, точно и понятно все это объяснил; его доклад «Тройная игра открытого кода» [www.redhat.com/magazine/001nov04/features/tripleplay/] прекрасно описывает эту ситуацию. Именно пользователи определяют направление движения проекта любого программного продукта, неважно, в проприетарной сфере или нет. Стоит ли красть чьи-то идеи и реализовывать их, если авторы сами могут участвовать в проекте на равных основаниях, сэкономив ваши деньги?

LXF: Это извечная и классическая проблема разработки программного продукта. Возвращаясь к разговору о сообществе Fedora: когда мы в последний раз общались с ним в 2006 году, оно было достаточно аморфным, и по большей части им руководили из Red Hat. С тех пор сообщество сильно выросло, что заметно по активности в списках рассылки, блогах и wiki разработчиков и т.д.

ДжЭ: Оно очень сильно выросло, и в большой степени благодарить за это стоит принятую модель управления. То, что небольшие группы по интересам знают, что их слово будет учитываться, естественно, без ущерба общей картине проекта. Сначала нас всех волновала реформа всех частей KDE в Fedora, но сообщество, похоже, преспокойно это съело: они справились с задачей. Предлагая поддержку группам, прокладывающим дорогу, мы сможем справиться с чем угодно. Конечно же, нам очень помогло, что мы усадили у себя часть разработчиков KDE! Все отлично сработано. Теперь у нас есть KDE Live CD, и все абсолютно современное. Последние разработки в Gnome нас тоже впечатляют: акроним для Gnome Online Desktop – GOD – я его просто обожаю!

LXF: На это могут поступить жалобы! Лично я использую KDE на своей основной рабочей машине в редакции. Итак, вопрос, который напрашивается: Fedora 8 выйдет вовремя? Или вы решили отодвинуть дату выхода до Нового Года, чтобы включить KDE 4?

ДжЭ: Мы планируем релиз на начало ноября, появится KDE 4 или нет. И пока что мы не выбиваемся из графика. Во всех релизах, кроме Fedora 6, мы свои планы выполнили. В дальнейшем мы собираемся делать релизы на Майские праздники [в Великобритании отмечается Майский праздник – 1 мая, он носит традиционно-обрядовый характер, – *прим. пер.*] и на Хэллоуин, что, похоже, соответствует общепринятым графикам выхода релизов; намного лучше иметь скользящий график.

В последнее время мы много внимания уделяли QA – контролю качества. Уилл Вудс [Will Woods], глава Fedora QA, проделал фантастическую работу. Даже в Fedora 7 кое-что, так сказать, проскользнуло под радаром, и мы хотим ограничить подобные вещи в будущем.

LXF: Ну, исправления делаются достаточно быстро. С некоторыми дистрибутивами бывает, что установишь его в день выхода релиза, а через неделю выясняется, что надо обновить 200 пакетов. Налицо прогресс: 7-я уже менее глючная, чем 6-я.

ДжЭ: Fedora 6 была на самом деле очень хорошим релизом, все было учтено! Но нас никогда не удовлетворяет достигнутый статус-кво, нам постоянно надо покорять новые вершины, проводя тестирование и убеждаясь, что люди раньше получат доступ к этому, и что еще больше людей охвачено.

LXF: Что вы думаете о выборе фирмой Dell дистрибутива Ubuntu для заводской установки на свое оборудование? Вам не кажется, что Fedora следует нацелиться на раскрутку в этой области в 2008 году?

ДжЭ: Некая часть меня, конечно, говорит: «Да!», но более разумная часть советует: «Подождите!». То, что делает Ubuntu – замечательно, потому что мы все стремимся к тому, чтобы Linux на долгие годы стал предустановленной системой для покупателей компьютеров. Мы, конечно, не согласны с такими мелочами, как закрытые кодеки и драйверы, но ведь они поделятся с нами опытом, и мы решим эти проблемы тем способом, который считаем правильным. Мы работаем над такими вещами, как *Smolt*, профайлером оборудования. Мы хотим, чтобы у нас была возможность отслеживать статистику – для подхода к компаниям, выпускающим, например, беспроводные карты, чтобы убедить их открыть драйвера, опираясь на данные о числе пользователей такого дистрибутива, как Fedora. Выигрыш обоюдный: если пользователи Fedora обнаружат ошибку, например, в драйверах компаний, производящих беспроводные карты, мы можем сообщить об этом компании и даже помочь исправить эту ошибку.

Джек про Mugshot

«Все используют онлайн-сервисы; есть точка зрения, что дистрибутивам вроде Ubuntu надо самоопределиться...»

LXF: Я никогда не понимал, почему компании так настаивают на драйверах с закрытым кодом. Если компания создала какой-нибудь адаптер PCMCIA с поддержкой Linux, любой пользователь Linux с радостью его купит, потому что оно «просто работает».

ДжЭ: Обычная отговорка производителей оборудования – «у нас нет инженерных ресурсов, чтобы обеспечивать поддержку», однако это ошибочный аргумент, потому что им и не нужны ресурсы: просто дайте людям нужные инструменты, и они все сделают за вас. Очень грустно, но это лишний раз демонстрирует, что многие бизнесмены в компьютерной сфере по-прежнему не имеют ни малейшего представления об идеалах Open Source. Вопрос об ИС их пугает. Sun как раз сейчас столкнулась с этой проблемой: одному из их проектов нужно использовать видеодрайверы, код которых закрыт производителями, а те не желают даже пытаться представить себе преимущества открытия кода. Проблема курицы и яйца. Будем надеяться, что благодаря программам типа профайлера оборудования *Smolt* в Fedora мы подберем к ним ключик, и то, что они считают политическим аспектом Open Source, станет для них меньшей проблемой, когда они оценят выгоды.

AMD объявили на встрече в Red Hat, что они готовы открыть некоторые из своих драйверов. Сообщество Fedora предложило им помощь, и, похоже, это предложение встречено хорошо. Мантрой Fedora всегда было «Делать все правильно», и из-за этого у нас возникают как бы личные проблемы, когда дело доходит до Ubuntu. Пользователи загружают этот дистрибутив, и хотя он «просто работает», они не узнают ничего нового о том, как много может сделать сообщество Linux, чтобы изменить ситуацию с закрытым кодом. Это не то что вредно, но, по-моему, способствует тому, что новые люди в мире Linux принимают все таким, как оно есть…

LXF: Однако пользовательская база Linux за последние три-четыре года сильно изменилась. В прошлом это был сознательный выбор, политический или идеологический: люди принимали концепцию свободного программного обеспечения; а теперь, в какой-то мере из-за роста популярности, об этом уже не думают и просто пользуются Linux по причине удобства, или по каким-то техническим причинам. Примером может служить рост популярности Linux на рынке домашнего аудио — журналы вроде LXF поставляют на своих дисках Ubuntu исключительно в качестве платформы, на которой их читатели могут работать с открытыми приложениями по созданию аудио, которых нет у Windows или OS X.

ДжЭ: Fedora уже два года работает над платформой Mugshot. Следующим шагом в компьютерных привычках пользователей станет рабочий стол онлайн. Все пользуются Gmail, Facebook, RSS, всеми этими онлайн-сервисами; но есть такая точка зрения, что дистрибутивам, подобным Ubuntu, надо самоопределиться, и в области онлайн-десктопов Ubuntu отстает от Fedora. Мы не пытаемся конкурировать, и не занимаемся инновациями ради инноваций: при наличии разных дистрибутивов, концентрирующих внимание на разных сферах, Linux в целом сможет предложить такой выбор, который целиком и полностью превзойдет коммерческие предложения.

Я не могу говорить за Хавока Пеннингтона [Havoc Pennington, ответственный за поддержку сайта социальной сети Mugshot http://mugshot.org, объединяющего контент из MySpace, YouTube, Facebook, Flickr, блогов и прочего], но я убежден, что мы просто превзошли самих себя с помощью Mugshot: мы – впереди всех, в этой области до нас не дотянуться.

Стремление Fedora быть впереди важно не только для хакеров: гораздо важнее то, что оно ведет к изменениям общества. Потому-то мы и мы любим работать с Creative Commons – там продвигают реформу в сфере авторских прав: произведенные изменения коснутся всех, а не только любителей потрошить компьютеры. Компьютерщики старой школы, например, Брайан Белендорф [Brian Behlendorf, руководитель технологического отдела CollabNet] всегда ищут что-то масштабное, так что мы надеемся быть на переднем плане всего происходящего, чтобы содействовать «правильным делам». Мы хотим давать решение как можно большему количеству пользователей; люди должны обустраивать свой дистрибутив так, как им это надо.

LXF: Мы тоже пытаемся выяснить, какова наша читательская аудитория. Кто они, наши читатели – ИТ-профессионалы? Пользователи? Разработчики? В итоге мы осознали, что это не столь важно: мы просто даем объяснения и показываем, что надо делать. И не важно, используют ли они это все в бизнесе, или им просто интересно этим заниматься. Важно, что ни публикации, ни дистрибутивы Linux не пытаются переосмыслить ситуацию или классифицировать свою аудиторию.

ДжЭ: Каждый человек уникален, и каждая компания уникальна. Уникальность движения свободного ПО в том, что оно лучше соответствует потребностям людей и обращает больше внимания на их пожелания, чем закрытые проприетарные программы, которые «почти соответствуют». Хотя выбор и важен, движению Open Source нужно единство, чтобы не дублировать усилия; порой бывает полезно заново изобрести велосипед – чтоб он стал лучше.

LXF: Какие главные трудности ждут Fedora в следующем году?

ДжЭ: Наверное, самая большая – самоопределение. Что такое Fedora? На данный момент мы пребываем в некой сумеречной зоне. Мы – дистрибутив? Или – платформа для множества других программ? Или система для создания приложений?

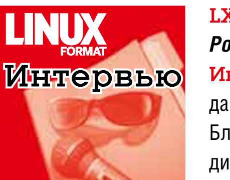
LXF: И каков же ответ?

ДжЭ: Многое для многих. Полем битвы в следующем году будет дифференциация. В чем мы будем непохожи на других? Недавно я читал книгу под названием «Взгляд на мир с точки зрения футбола [How Soccer Explains The World]» (HarperCollins, ISBN 0-066-21234-0); так вот в ней тема самоопределения по Франклину Фоэру [Franklin Foer] заключается в том, что мы пришли к некой точке цивилизованного племенного союза: можно легко войти в него и так же легко выйти. Вот это и происходит с Linux: неделю вы пользуетесь Fedora, еще неделю – Ubuntu, и еще неделю – чем-то еще. По сути, все они одинаковы, все ищут конфигурационные файлы в одном и том же месте, я ищу свои настройки, и они все там же… А людям нужно больше; и я полагаю, что наш долг – дать им это большее. Есть что-то интересное в том конвейере, который заново определит и брэнд Fedora, и ее место, и, возможно, полностью все изменит. Меня это не беспокоит: когда появился Gentoo, был прилив общего энтузиазма. В жизни все повторяется. LXF

PostgreSQL идет к людям

PostgreSQL – удивительно мощная и надежная система хранения данных. Молодая фирма POSTGRESMEN специализируется на ее укрощении и приручении. **Евгений Балдин** решил разобраться, как это делается, а помогает ему **Иван Золотухин** – ведущий разработчик и соучредитель компании POSTGRESMEN.





LXF: Чем был вызван ваш интерес к *PostgreSQL*?

Иван Золотухин: Все определила среда, из которой я вышел – астрономы МГУ. Благодаря Олегу Бартунову (LXF85) среди астрономов постоянно появляются сильные программисты, которые интересуются *PostgreSQL*. Кто-то начинает писать модули для *PostgreSQL*. В качестве примера можно привести *Q3C* Сергея Копосова (LXF89) или *pgSphere*, в создании которой участвовал Игорь Чилингарян. Кто-то делает крупные системы на его основе – это и был мой путь. Довольно быстро стало понятно, что My*SQL* не удовлетворяет требованиям, которые предъявляют к СУБД серьезные проекты. С тех пор наша команда, которая сейчас составляет ядро компании «Постгресмен», создала несколько известных систем на основе *PostgreSQL* и ни разу в нем не разочаровалась.

LXF: Как и зачем возникла ваша компания?

ИЗ: Идея создания *PostgreSQL*-компании, можно сказать, витала в воздухе. Мы давно обсуждали с Олегом перспективы и возможности ее открытия. Необходимость в этом, очевидно, назрела. К сожалению, пользователи *PostgreSQL* в России довольно разобщены, и неприлично мало внимания уделяется популяризации на рынке корпоративного ПО такой достойной СУБД, как *PostgreSQL*. Многие компании просто не могут заключать договоры с частными лицами на оказание услуг, связанных с *PostgreSQL* – спрос не находит предложения по причине юридических или организационных сложностей.

Чтобы исправить положение, а также удовлетворить растущий спрос на *PostgreSQL*-решения и предоставить качественную и квалифицированную поддержку корпоративного уровня, и была создана компания «Постгресмен». Мы стремимся не только извлекать прибыль, но и помогать развитию и распространению *PostgreSQL* в России.

LXF: Каковы масштабы деятельности вашей фирмы?

ИЗ: В компании на постоянной основе работают несколько сотрудников. Кроме того, по мере необходимости мы привлекаем внештатных консультантов.

LXF: Чем вы лучше других?

ИЗ: Все очень просто. «Постгресмен» – первая и пока единственная компания, предоставляющая русскоязычную 24/7/365-поддержку *PostgreSQL* в России и странах ближнего зарубежья. Наши специалисты являются настоящими экспертами в своей области. Клиенты нам доверяют, а мы не обманываем их ожидания. Видимо, дело в этом.

LXF: Чем же так привлекателен *PostgreSQL*?

ИЗ: О технических преимуществах *PostgreSQL* можно говорить долго. Если совсем кратко, то эта система способна конкурировать с другими СУБД как в сегменте «легких» баз данных, так и на огромных объемах и нагрузках, не только не уступая, но зачастую и превосходя коммерческие СУБД по скорости и надежности. Кроме того, в *PostgreSQL* реализовано множество технологий, которые просто отсутствуют во всех других СУБД. Некоторые из них способны сделать *PostgreSQL* лидером по использованию в соответствующих отраслях ИТ. Наших клиентов привлекает и то, что *PostgreSQL* общепризнанно является самой развитой из бесплатных систем управления базами данных.

LXF: Недавно в *PostgreSQL* 8.3 был официально включен модуль полнотекстового поиска *tsearch2*, созданный Олегом Бартуновым и Федором Сигаевым (LXF86). Насколько это важно для пользователей?

ИЗ: *Tsearch2* – уникальный модуль, с огромными возможностями и максимально гибкой конфигурацией. Благодаря всем этим качествам он является самым популярным «контрибом» в *PostgreSQL* вот уже долгое время. Сейчас он по просьбам пользователей совершенно заслуженно был перенесен в ядро СУБД, став самым большим изменением исходных текстов *PostgreSQL* за всю его 11-летнюю историю.

В нашей деятельности *tsearch2* имеет огромное значение: практически все системы, созданные нами, используют полнотекстовый поиск. Непосредственно поисковиками мы не занимаемся, но благодаря отличной масштабируемости *tsearch2* на его основе можно делать и такие системы. Собственно, они и существуют – например, поиск по документам в сети МГУ или тематический поисковик **pgSQL.ru**.

Миграция полнотекстового поиска в ядро *PostgreSQL*, кстати, даст возможность пользователям виртуальных хостингов использовать в своих приложениях полноценный поиск, не попадая в зависимость от провайдера и того, какие модули к *PostgreSQL* он устанавливает на своих серверах. Это очень важный момент, который непременно хочется отметить.

LXF: Что можно сказать про создание кластеров на *PostgreSQL*?

ИЗ: *PostgreSQL*-кластеры просто необходимы там, где имеются серьезные нагрузки на СУБД или требуется максимальная отказоустойчивость системы. Мы всегда рекомендуем своим клиентам использовать масштабирование на основе принципа scale-out за счет одинаковых относительно недорогих серверов, в противоположность принципу scale-in, когда деньги вкладываются в усовершенствование единственного *PostgreSQL*-сервера.

Slony, будучи самым популярным решением по master-slave репликации *PostgreSQL*, не является единственным. Есть и мультимастер-репликация, например, *PgCluster* или *pgpool-II*, существуют альтернативные открытые решения от компании Skype, которые тоже представляют большой интерес в задачах масштабирования *PostgreSQL*.

К сожалению, жанр интервью совсем не подходит даже для беглого обзора систем репликации и масштабирования; это, кстати, одна из самых популярных тем, которые интересуют слушателей наших семинаров. Благо, нам есть что рассказать.

LXF: Что можно сказать про интеграцию *PostgreSQL* и 1С? Насколько это сейчас востребовано?

ИЗ: Интерес к такой интеграции стремительно растет, уже появились первые компании, которые полноценно используют Linux/*PostgreSQL*-версию «1С: Предприятие». На профильных форумах сейчас можно наблюдать множество вопросов на эту тему. И интерес к *PostgreSQL* в сообществе 1С в ближайшем будущем будет только увеличиваться.

Пока не так много 1С-компаний предоставляют услуги перехода на *PostgreSQL*-версию. С другой стороны, появляются небольшие компании, которые специализируются исключительно на переводе системы 1С на Linux. Словом, рынок только формируется, но у пользователей уже появляются возможности для выбора. Кто-то, конечно, осуществляет миграцию самостоятельно. Пользователям, имеющим навыки чтения документации, это вполне по силам. Кроме того, мы проводим семинары на эти темы, и они пользуются неизменным спросом.

Что касается интерфейса *1C-PostgreSQL*, то, конечно, в нем не все идеально. Но мы надеемся, что в последующих версиях ситуация будет улучшаться, ведь у *PostgreSQL* огромные возможности – осталось только воспользоваться ими.

LXF: А что *PostgreSQL* может дать, скажем так, обычному пользователю, вооруженному *OpenOffice.org*?

ИЗ: Во многих компаниях возникает, например, следующая ситуация: какие-либо данные готовятся «операторами ПК» с использованием электронных таблиц, а затем отдаются программистам для автоматизированной обработки, когда возможностей таблиц вроде Excel перестает хватать. С помощью *OpenOffice.org* и *PostgreSQL* возможно абсолютно прозрачно для «операторов ПК» сохранять информацию сразу в реляционной базе данных, предоставляя тем самым *SQL*-интерфейс доступа к ним. В нескольких случаях, с которыми мы сталкивались, такой подход оптимизировал работу в разы, при практически нулевых затратах. Кроме того, мы симпатизируем *OpenOffice.org* и не упускаем возможности показать, что он мало чем уступает коммерческим аналогам, будучи при этом абсолютно бесплатным.

LXF: Чего вам не хватает в *PostgreSQL*?

ИЗ: Практически все, чего сейчас не хватает *PostgreSQL*, перечислено в его TODO-листе. Конечно, у нас есть специфические пожелания, вызванные потребностями клиентов. Это, например, ускорение префиксного поиска и поиска *n* точек, ближайших к заданной, в двумерном случае.

К недостаткам *PostgreSQL* я бы отнес и относительно слабую поддержку этой СУБД со стороны хостинг-провайдеров в настоящее время. Однако мы, тем не менее, наблюдаем положительную динамику в этом отношении.

LXF: Что бы вы пожелали миру свободного ПО?

ИЗ: Нам бы очень хотелось, чтобы в российских компаниях более серьезно относились к бесплатному ПО, которое зачастую превосходит по качеству коммерческие аналоги. Грядет вступление России в ВТО, и нелицензионные программы должны уйти в прошлое. На наш взгляд, в настоящее время на рынке корпоративного ПО наблюдается острая нехватка популяризаторов открытых продуктов, и мы будем только рады, если компании, подобные нашей, будут появляться в других отраслях свободного ПО. LXF

Что за штука CMake?

Эй, разработчики! Все знают, что Make – программа путаная и обветшалая; **Пол Хадсон** за ручку сведет нас в счастливую страну CMake!

» *Make* – мой архивражина! Ради Бога, скажите: вы ведь избавите меня от нее?
Я – нет, а вот команда *CMake* собирается! Хотя и это еще не факт, ведь вытеснить *Make* пытались неоднократно, но разве только *Ant* добилась умеренного успеха. Несомненно, *Make* будет с нами еще долгие годы, даже если *CMake* победит.

» Беда-а-а! Ненавижу ее манеру различать табуляцию и пробелы – *Make* до того неудобна, даже для бывалых линуксоидов!
Я согласен – и, по-моему, большинство Linux-программистов тоже. Но при всех достоинствах *CMake*, *Make* никуда не денется: уж больно много проектов с ней связано.

» Выходит, у *Cmake* есть достоинства? Да говорите же!
OK – начнем с начала. «C» в названии проекта означает «кроссплатформенная» и символизирует отличие от *Make*: программа не выполняет компиляцию, а лишь создает файлы, необходимые для компиляции на конкретной системе. Предельно упрощая, *CMake* – лишь тонкая оболочка вокруг *Make*, призванная скрыть уродство последней. Главное же достоинство *CMake* – переносимость и на OS X, и на Windows: там, где не будет *Make*, она воспользуется «родным» инструментом другой системы.

» Постойте... так *Make* все-таки придется оставить?
Make или другую эквивалентную программу для сборки ПО. Причем программисты одной команды не обязаны использовать одну и ту же утилиту: работая с *Qt* под Linux, вы выберете *Qmake*, а ваши друзья в Windows предпочтут стандартные средства *Visual Studio*. Вы можете использовать общие файлы конфигурации *CMake*, то есть все сделанные вами изменения тут же появятся на всех платформах. Более того, команде *CMake* незачем писать подключаемые модули для поддержки разных IDE: ведь все программы останутся при своих «родных» форматах сборки, вмешиваться в которые *CMake* не будет.

» Значит, даже при победе *CMake* будет не обойтись без *Make*? Ну и где тут особое улучшение?
Да, пользоваться *Make* придется, но только косвенно – *CMake* создаст все необходимые файлы, избавив вас от мучений. При запуске *CMake* нужные файлы генерируются и хранятся в директории **/source**, и обычно вам даже заглядывать туда не надо.

» Ладно, одна причина для обновления есть, но этого явно мало. Зачем мне суетиться?
Мало причин? Вот вам четыре миллиона строк причин: на *CMake* переключается KDE 4. Намаявшись с GNU Autotools, разработчики KDE потратили кучу времени на поиски инструментов-альтернатив и в итоге выбрали *CMake* – как наиболее гибкую и модифицируемую. Именно модифицируемость сыграла решающую роль: ведь KDE – громадный проект со специфичными требованиями. Команды KDE и *CMake* наладили тесное взаимодействие, *CMake* значительно усовершенствовалась, стала еще надежнее и стабильнее, а главное – получила мощнейшую поддержку!

» И *CMake* справляется с большими проектами?
Разумеется! *CMake* прекрасно взаимодействует с KDevelop, и способна собрать любой компонент KDE – от *kioslave* до *Kpart*. Вы получаете неограниченный контроль над системой зависимостей, и я буду очень удивлен, если чем-то окажется нельзя управлять...

» Судя по вашим словам, *CMake* используется только в KDE...
Вовсе нет, но не стану отрицать, что *CMake* находится в самом начале пути. Другой известный Linux-проект, уже переключившийся на *CMake* – *Scribus*; Питер Линнелл [Peter Linnell], web-мастер *Scribus*, сказал: «Установив *CMake*, я за 10 минут во все вник и встроил файлы *CMake* в наш исходный код. Красота!». Кроме KDE и *Scribus*, *CMake* поддерживает *SWIG* и *LaTeX*, а это уже путь к всеобщему признанию.

> «Самый известный Linux-проект, перешедший на CMake – настольная издательская система Scribus».

» Трудно ли настроить *CMake*?
Ответ зависит от того, насколько сложной вам казалась *Make*. Скажем прямо: простой программой *Make* не считал никто. Команды *CMake* напоминают легкий программный код, пользоваться которым сравнительно несложно, если у вас есть минимальный опыт программирования – хотя бы написание простенького JavaScript. Многое сделано для автоматизации процесса, перевода рутинной работы на задний план, чтобы вместо составления файлов заниматься собственно программированием.

» Вы упоминали, что *CMake* – кроссплатформенная программа; на каких же платформах она работает в действительности?
Известные системы поддерживаются все: Linux, Windows, OS X, Solaris, FreeBSD, HP-UX, IRIX, BeOS и

QNX; любая система, «признающая» *Make*, пригодна для *CMake*. Куда важнее, что программа совместима с *KDevelop*, *Xcode* и *Visual Studio*, а ведь они-то и интересуют профессиональных программистов! Естественно, поддержка улучшается с каждой новой версией.



А в чем преимущество *CMake* по сравнению, скажем, с *Ant*?

Ant – полная, радикальная замена *Make*. Радикальность здесь и преимущество, и недостаток: если предпочесть *Ant*, то и ОС, и все остальное ПО должны поддерживать ее. Что касается ОС, то *Ant* написана на Java, и проблемы с системой маловероятны.



Подозрительно: «Java» и «нет проблем» в одном предложении встречаются редко...

И то правда, но допустим, что все работает гладко. Реальная проблема *Ant* состоит в том, что ее должна поддерживать среда разработки – нужно заботиться об установке нужных модулей расширения вашего ПО, или надеяться на готовую поддержку *Ant*. Тот факт, что *CMake* не отменяет *Make* – явное преимущество: ваш инструментарий остается тем же.



И вы обещаете, что мне не придется читать файлы, созданные *CMake*?

Обещаем!



А можно ли при необходимости «подправить» программу?

Программа выпускается под BSD-подобной лицензией, допускающей неограниченное использование, в том числе и в составе коммерческих продуктов (разумеется, при соответствующем оформлении). Изменяйте исходный код на здоровье! Если боязно трогать исходник, расширить программу можно и без этого – она действительно очень гибкая.



ОК, уже хочу попробовать. А если я увязну, куда обратиться за дополнительными сведениями?

Сайт онлайн-документации *CMake* – www.cmake.org/HTML/Documentation.html, но для типичных вопросов предусмотрен www.cmake.org/Wiki/CMake_FAQ. Есть прекрасная статья о причинах перехода KDE 4 на *CMake*: http://lwn.net/Articles/188693.

Если вы завязли здорово, или решили использовать *CMake* в серьезной работе, обратитесь к книге *Mastering CMake*: www.kitware.com/products/CMakebook.html. Много информации о *CMake* можно найти на *CMake* wiki (www.cmake.org/Wiki/CMake), но мы особо рекомендуем руководство по быстрому ознакомлению (www.CMake.org/HTML/Examples.html), учебник *CMake*/Eclipse (www.CMake.org/Wiki/CMake:Eclipse), и азбуку Ctest, инструмента тестирования *CMake* (www.CMake.org/Wiki/CMake_Testing_With_CTest). Как видите, чтения хватает, но – вспомните эту возню с освоением *Make*, и вы поймете, что не все так плохо! LXF

Прими участие

Сейте доброе и вечное – объясните другим, почему вы влюблены в Linux! Вот пять путей для успешной проповеди Open Source.



Любите ли вы свободное ПО? А много ли времени вы тратите на распространение этой идеи среди друзей, семьи и соседей, чтобы они тоже его полюбили? Доминирование Microsoft в компьютерной сфере медленно, но верно сходит на нет, а уж если мы все объединим усилия в продвижении решений с открытым исходным кодом, то грохот рушащейся монополии наполнит уши Стива Балмера быстрее, чем вы успеете произнести: «Разработчики, Разработчики, Разработчики!»

Мы не бросим вас в одиночку искать ресурсы в Интернете: с гордостью представляем наши пять лучших советов для содействия распространению «Слова Линуксова» и свободного программного обеспечения – читать обязательно!

1 Создай LUG

› **ScotLUG в прошлом году на Пивном фестивале в Пейсли. А в вашей LUG пиво любят?**

Ежемесячная [в английской версии журнала, – *прим. перев.*] страница LUG [Linux User Group – Группа пользователей Linux] в **LXF** наглядно показывает, что групп пользователей Linux в мире хватает, но почему бы не создать и собственное сообщество – особенно если вы занимаете уникальную нишу? Например, если ваша группа пользователей помогает новичкам решать проблемы роста, это уже отличает вас от большинства других групп! Организовывать LUGи на удивление легко, потому что они, по большей части, создаются на неофициальных встречах в районе «базирования» ее членов. Не надо непременно снимать конференц-зал – ничуть не хуже просто пойти в бар, тем более, что там теперь не накурено [в английской версии реальности, – *прим. ред.*]. Вдобавок в баре каждый может выбрать себе напитки и угощение по вкусу.



Создав свою LUG, не забудьте ее зарегистрировать: **www.lug.ru** – неплохое место для начала, но также сообщите об этом нам в *Linux Format*: быть может, мы составим свою карту LUG'ов России. Существуют также рассылки LUGов и группы новостей, где вы должны опубликовать объявление о вашей новой LUG: очень важно хорошенько потрудиться над привлечением новых членов в первые несколько недель. Когда ваша LUG заработает, вам сразу полегчает, потому что появятся новые люди, и часть нагрузки ляжет на них, но в первое время за все отвечаете вы! Если вы приметесь запускать LUG с одним или двумя соратниками, вам будет намного проще.

Помните, что быть частью LUG – это больше, чем просто «ворковать» через кабель категории 6 на встречах. Вот несколько простых способов, которые помогут вам вызвать интерес к действу:

» **Раздайте листовки-приглашения** или оставьте их на досках объявлений в общественном месте. Библиотеки, колледжи и университеты – хорошее место для такого рода вещей, но сперва получите на это разрешение! Запись на странице раздела предстоящих событий корпоративной интрасети также отличный вид рекламы: весьма вероятно, что поклонники Linux есть среди ваших коллег в других отделах компании.

» **Поощряйте людей приносить необычное оборудование.** Суперкомпьютер Cray явно никто не притащит, но хакеры любят поиграть с новыми штучками – например, если кто-то сумел урвать iPhone через пару дней после его запуска в США, это уж точно вызовет неподдельный интерес!

» **Проводите праздники запуска.** Когда выйдут *Firefox* 3 и KDE 4, люди захотят их опробовать и отметить релиз, поэтому убедитесь, что LUG принимает в этом участие – выставьте ноутбук с новым программным продуктом, пиво и закусь, чтобы все расслабились и приятно провели время!

» **Развивайте другие интересы.** Большинство фанатов компьютеров интересуются научно-фантастическими фильмами, событиями в аст-

рономии или другими техническими вещами, и ваша LUG – отличный повод насладиться этим вместе.

» **Пригласите ораторов.** Большинство знаменитостей свободного ПО охотно выступят перед вашей LUG бесплатно, если вы покроете им расходы на поездку. Возможно, хотя бы один известный разработчик найдется в радиусе нескольких сотен миль от вашей LUG, и расходы вас не разорят.

» **Используйте местных специалистов.** Если не получается пригласить знаменитостей, не отчаивайтесь: есть шанс, что члены вашей группы достаточно грамотны, чтобы провести небольшой доклад о программах или методах, которыми они сами интересуются. Блиц-выступления, на которые отводится пять-десять минут, заставляют говорить интересно и по делу и вполне могут дать представление о теме.

» **Привлеките спонсоров.** Издательства вроде O'Reilly и Pearson, как правило, с удовольствием посещают заседания крупных LUG, либо с книгами, либо с авторами, готовыми продвигать свои книги путем семинаров. Свяжитесь с ними и узнайте, не смогут ли они вам помочь!

» **Специализируйтесь.** Вокруг вас существует много LUGов, они имеют множество сайтов, которые содержат чуть больше, чем даты встреч и списки адресов. Если членам вашей LUG в первую очередь интересна какая-то конкретная тема, например, «Linux на web-сервере», воспользуйтесь этим – загрузите Wiki на сайт и попросите каждого поделиться знаниями с миром!

2 Проведи День Установки

Классический способ проталкивания Linux в мир – предложение установить Linux кому-нибудь на компьютер. Вам это представляется сложным и трудоемким? Зря. Установить новейшие версии Ubuntu [да и других дистрибутивов, – *прим. перев.*] до смешного легко: вставляете диск, тыкаете несколько кнопок, а через 10 минут возвращаетесь на готовенькое. Поскольку Ubuntu работает и как установщик, и как Live CD, это дает людям шанс опробовать его на своих ноутбуках без перекраивания своей системы, а заодно и возможность убедиться, что их оборудование полностью поддерживается!

Но Праздник установки – Инсталфест – не то событие, что можно организовать за пару минут; если вы хотите выжать из него максимум, не мешает провести подготовку. Вот наш перечень важных пунктов:

» У вас есть ПК с уже установленным Linux? Прежде чем соглашаться поставить Linux на свой компьютер, люди почти всегда хотят сперва посмотреть его в действии – будьте готовы позволить им свободно экспериментировать с компьютером, чтобы они узнали все, что им требуется, до установки. Лучше всего здесь воспользоваться системой виртуализации, типа *VMware Server* [а также *Qemu*, *Virtualbox*, – прим. перев.], чтобы предоставить на пробу пять-шесть различных версий Linux, без необходимости следить за пятью-шестью ноутбуками!

» Предоставляйте личные учетные записи только с жестко ограниченными правами, чтобы никто случайно не грохнул ваш Linux. И не дай Бог, чтобы у вас с языка сорвалось слово "root"!

» Припасите несколько установочных дисков. Хотя установка Ubuntu на жесткий диск и известна быстротой, 20–30 минут все-таки вынь да положь; а если вас попросят установить Linux сразу несколько человек, диски должны быть наготове. Лучше всего иметь штуки три, плюс ISO-образы на ноутбуке с записывающим CD-приводом, на случай, если трех не хватит.

» Приведите с собой людей, способных оказать кому-то помощь, пока вы занимаетесь установками. Помните: если вы установили 50 систем, но лишь 20 и работают корректно, и их пользователи понимают, что делать дальше, это довольно скверный итог!

» Будьте готовы ответить на вопросы широкого круга интересующихся – пользователей Mac и Windows, специалистов по издательским системам и администраторов серверов, чародеев *Photoshop*'а и людей, которые думают, что *Internet Explorer* – это и есть Интернет, а также многих других. Удобно также иметь под рукой список эквивалентных программ: например, *Microsoft Office* -> *OpenOffice.org*.



Дни установки, проведенные LUG Йорка в местном университете в выходные 4–5 ноября прошлого года, имели большой успех.

» Разложите вокруг книги или повесьте плакаты, чтобы люди могли чем-нибудь заняться, пока ждут.

» Позаботьтесь, чтобы люди могли хоть что-то забрать с собой, даже если это просто печатный листок с основными инструкциями и контактным адресом электронной почты на случай, если у них возникнут вопросы. Возможно, не лишним будет обвешать их сияющую новизной установку Linux ссылками на популярные сайты, где им могут помочь.

» Если вы член LUG (или если вы выполнили наши рекомендации и создали свою собственную группу!), известите всех участников Инсталфеста о вашей следующей встрече. Неделя-другая на освоение новой системы – достаточное время, чтобы накопить приличный список вопросов, которые они смогут задать членам вашей LUG на очередном мероприятии. Разместите эти вопросы – обязательно с ответами! – на своей web-странице, чтобы другие люди также могли ими воспользоваться.

» Избегайте тона упертого фанатика свободного ПО! Если люди, например, зададут вопрос: «А Linux так же легко использовать, как OS X?», отвечайте им честно. Linux – прекрасная, свободная система со множеством преимуществ, и не надо бояться говорить о ее недостатках, еще и потому, что люди, пришедшие на Инсталфест, ждут правдивых ответов для принятия решения.

3 Раздавай CD

Лучший способ заинтересовать людей Linux'ом – это дать им его попробовать: те, кто продвинулся дальше первичного «а где у меня диск C:\?!», как правило, находят, что Linux – особенно в части *Firefox* и *OpenOffice.org* – весьма похож на Windows. К счастью, Linux распространяется бесплатно, а компакт-диски нынче так дешевы, что вы сможете одарить всю округу за какую-нибудь 1000 рублей.

Тут, конечно, встает вопрос, что записать на эти диски. Вот вам кое-какие идеи: »

По всему миру

Некоторые LUG – особенно в азиатских странах, таких как Таиланд, где принятию Linux в настоящее время способствуют местные власти – могли бы извлекать прибыль, став агентами по подписке и/или распространителями нашего журнала в тех районах мира, где *Linux Format* не так доступен в газетных киосках. Пока вы читаете эту статью, мы обдумываем, как помочь сделать это сколь можно просто: если вашу LUG или университетскую группу вне Англосферы заинтересует участие в подобной схеме, пожалуйста, обратитесь к нам – мы хотели бы узнать у вас способы доставки тиража *Linux Format* к новой аудитории.



›› Хорошее начало – полный дистрибутив Linux, вроде Mint Linux. Мы поместили ISO Mint Linux Light на DVD в LXF98, можете скопировать его оттуда и дублировать по мере надобности.

›› Альтернатива – скачать новый дистрибутив Linux из Интернета и употребить его как базовую систему. Но убедитесь, что он прост в использовании!

›› Документация: если на диске остается место, разместите там свободные учебники и HowTo (их везде полным-полно), чтобы помочь людям начать. Кроме того, можно как минимум указать им полезные сайты.

›› Если у вас диски большого объема, такие как DVD, можно записать несколько версий Linux, или улучшить их, добавив пакеты программного обеспечения, которое люди наверняка захотят использовать. Дополнительную информацию см. на стр. 31 в LXF97.

›› Хорошо бы ваш диск выглядел профессионально: если вы сделаете наклейки на дисках – уже неплохо; если вложите их в бумажные конверты – еще лучше. В идеале, люди будут использовать диски долгое время, а может, даже поделятся ими с другими людьми! Помните, что пользователи старых компьютеров могут не иметь DVD-привода.

›› Если вы рветесь продвигать именно Ubuntu, можно получить бесплатные диски для распространения, в привлекательных фирменных упаковках – см. www.ubuntu.com/getubuntu/shipit-faq.

4 Устанавливай свободное ПО

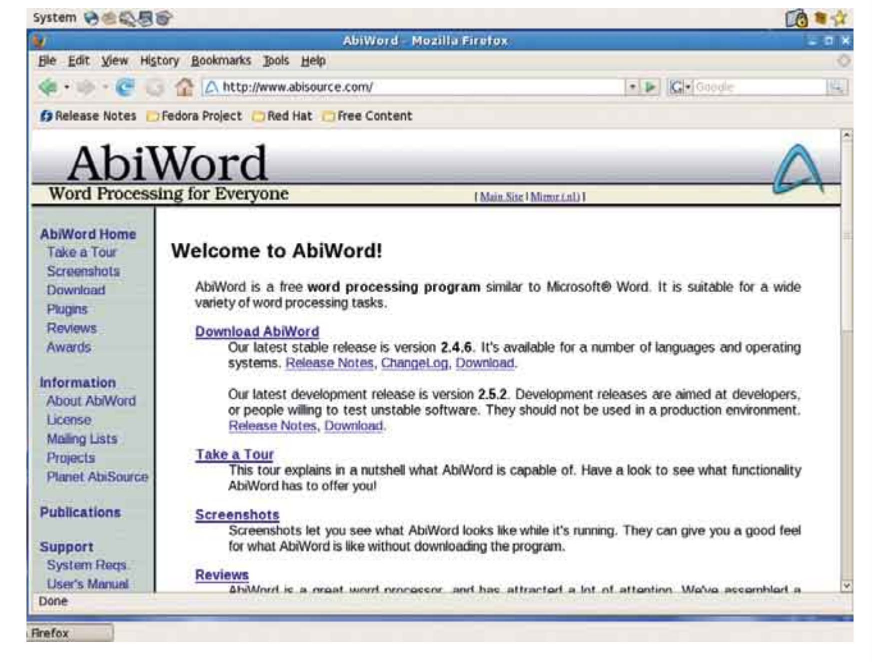
Для содействия Open Source вовсе не обязательно гнать всех знакомых на Linux – на самом деле, под Windows тоже можно добиваться свободы народов от программных монополий. Миллионы людей страдают от таких программ, как *Microsoft Works* или инфицированные программами-шпионами версии *Internet Explorer*, и они почти наверняка не в курсе, что на свете есть *OpenOffice.org* и *Firefox*. И тут являетесь вы и помогаете им перейти на свободные программы прямо в их же операционной системе! Преимущество здесь в том, что пользователь остается в комфортной родной среде (незачем, например, разбираться, как работает Gnome), а кроме того – созрев, наконец, для перехода на Linux, они будут уже знакомы с основными программами.

Вот хороший список ПО, которое им понравится:

›› **Офис:** *OpenOffice.org* и *AbiWord* – отличная альтернатива для большинства пользователей, хотя вы, возможно, типовой сборке *OpenOffice.org* предпочтете *Oxygen Office*, поскольку к нему прилагается больше шаблонов [российским пользователям стоит присмотреться к ООо Pro от Инфра-Ресурс, – *прим. перев.*].

›› **Web-браузеры:** очевидный выбор – *Firefox*, но это не значит, что вы должны ограничиться одной его установкой. Например, многие люди не догадаются о вкладках, пока вы им их не покажете.

›› **Электронная почта:** Если *Firefox* – стандарт де-факто для браузера, то *Thunderbird* – выбор по умолчанию среди почтовых клиентов: он



› *AbiWord* очень быстр, совместим с *Microsoft Word* и абсолютно бесплатен – истинная жемчужина в мире Open Source. www.abisource.com.

Амнистия Windows

Один из основных факторов, сдерживающих принятие Linux многими пользователями ПК, заключается в непонимании того, что Windows и Linux могут работать бок о бок. Хорошим способом заявить о наличии LUG людям в вашем районе может быть проведение «Дня амнистии Windows»: попросите людей принести свои ПК, и пусть ваш гуру Linux поможет им «расщепить» диск, сделав загрузку двойной – с Windows и Linux одновременно, или покажет, как загрузиться с Live CD или USB-накопителя.

LXF рассудил, что эта идея добавляет узнаваемости, запоминаемости и смысла не слишком-то привлекательным без такой амнистии «Дням установки Linux»: если их, скажем, организовать на национальной или международной основе, да найти спонсоров среди поставщиков дистрибутивов Linux – подозреваем, что реклама будет отменная…



быстр, легок и очень, очень мил. Великолепный шанс убедить людей, что бесплатность приложений с открытым исходным кодом не мешает им красиво выглядеть!

» **Манипуляция с графикой:** *GIMP* – прекрасный инструмент для продвинутых пользователей, но есть прекрасная программа (к сожалению, только для Windows) *Paint.NET* – это свободный инструмент, хотя частично и поддерживается Microsoft!

» **DTP:** *Scribus* доступен на нескольких платформах, причем его интерфейс на основе *Qt* повсюду одинаков. Хорошо и то, что *Scribus* суперлегок [ну-ну!..] в обучении и использовании [...ну-ну-2! Ведь сами же просили: честно о недостатках, – *прим.верстки*.]!

» **Игры:** есть куча игр для Linux, и многие из них прекрасно работают и под Linux, и под Windows. Игры, написанные для KDE или Gnome, вероятно, не заработают, но их процент невелик. Безопасно все, что использует SDL (Simple DirectMedia Layer), поскольку SDL работает на нескольких платформах, включая Windows и OS X.



5 Расширяй круг читателей LXF!

А уж самое доброе, что вы можете сделать – это предложить людям почитать свою подшивку номеров *Linux Format*: мы всегда пытаемся сочетать статьи, обзоры и новости для людей с разными уровнями подготовки, и если у вас завалялась стопка старых выпусков, вашим друзьям она может показаться информационным кладом – будет дополнительная выгода от ваших покупок! И, хотя Новый год уже прошел, подписка продолжается – преподнесите своим дорогим и любимым подарок безо всякого повода! LXF

ВАС ПРИГЛАШАЮТ!

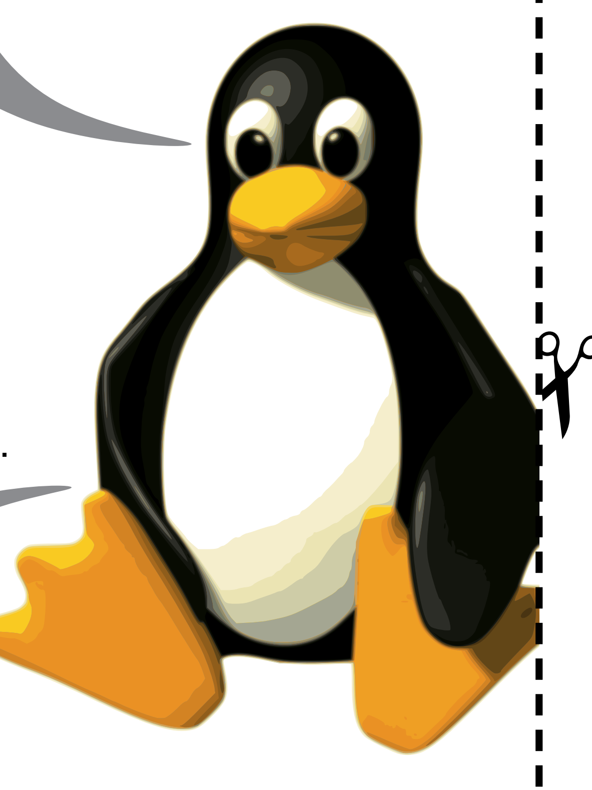
....

проводит встречу LINUX USER GROUP (LUG)

адрес:..

дата:...время:.................

LUG –
это сообщество пользователей
для обмена знаниями, опробования нового
ПО, помощи друг другу и просто для удовольствия.
Неважно, прожженый ли Вы линуксоид, или Вам
просто любопытно узнать, что может Linux –
мы ждем Вас!



› **Это приглашение размещено на нашем DVD в директории Magazine/Invite как документ *Scribus* (а также в формате PDF), который можно самостоятельно отредактировать.**

Горячие темы

Что думают разработчики дистрибутивов по поводу самых острых вопросов в сегодняшнем мире Linux? Майк Сондерс и Ник Вейч спровоцировали дебаты на LinuxWorld 2007...

Когда Линус Торвальдс [Linus Torvalds] негативно высказывается о Gnome, или Ричард Столлмэн [Richard Stallman] разражается грозной тирадой против проприетарного ПО, мы ловим каждое их слово, откликаясь обширными дискуссиями на форумах в сети. В конце концов, они главные фигуры в мире свободного ПО, и говорят вещи, способные радикально изменить судьбу нашей любимой ОС.

Но, пожалуй, самой важной и, одновременно, не любящей разбрасываться словами группой являются сами разработчики дистрибутивов – те программисты, что неустанно куют крутое программное обеспечение для наших компьютеров. Они часто видят проблему в ином свете, чем топ-менеджмент, потому что больше заняты программированием, тестированием и доработкой, чем разговорами.

Поэтому в этом году на LinuxWorld в Сан-Франциско, мы старались уделить больше времени разработчикам: мы пообщались с двумя хакерами из Gentoo и одним разработчиком Debian, выставив на обсуждение пару-тройку важнейших проблем мира Linux. Что они думают по поводу заявления Microsoft о патентах, взлете Ubuntu, а также отсутствия универсальной графической среды?



ДОН АРМСТРОНГ [DON ARMSTRONG]

Разработчик Debian по нескольким направлениям: поддержание пакетов, отслеживание ошибок и правовые вопросы, стоящие перед проектом. Его сайт www.donarmstrong.com.



КРИС ДЖАНЕЛЛОНИ [CHRIS GIANELLONI]

Лидер проекта Gentoo Release Engineering. Пришел в Gentoo в 2003, создает ebuild'ы для Игр.



МАЙК ФРАЙСИНГЕР [MIKE FRYSINGER]

Работает в Совете Gentoo и занимается инструментарием разработчика (GCC и т.д.). Также следит за портами для не-x86 архитектур, подобных ARM и S390.



› Не думайте, что здесь так уж пусто: правду сказать, мы всех отсюда выкурили, заведя беседу о форматах пакетов и о демократии…

1. Microsoft и патенты

В мае Microsoft заявила, что открытое программное обеспечение, в том числе ядро Linux, нарушает 235 их патентов. Это вызвало мощный резонанс в сообществе, особенно учитывая ситуацию вокруг SCO. Но такая ли это большая проблема?

Крис Джанеллони: Я думаю, что это полная чушь, пока не предъявлено что-то осязаемое: «Вот код, а вот доказательство».

Дон Армстронг: Я думаю, все знали, что у них есть 300 патентов, которые имели какого-то рода влияние на некоторые аспекты разработки Open Source. Но пока они не скажут: «Вот 300 патентов, которые нарушены», никто даже представляет, о чем речь.

И если честно, никто не собирается шарить среди их патентов, чтобы выяснить, что они из себя представляют, потому что никто не хочет отвечать за ущерб, или за незнание того, что у них там за дурацкие патенты. Поэтому до тех пор, пока они действительно не начнут отслеживать нарушения патентов... а они не пойдут против Debian или Gentoo, потому что с нас взять-то нечего! Я думаю, что наше совместное финансирование сможет покрыть от силы один день работы их адвокатов!

Довольно бессмысленно с их стороны нас преследовать. Пока они не найдут кого-то с деньгами, это действительно не будет иметь смысла.

LXF: Но какие последствия это может возыметь? Если Microsoft придет и скажет: «Эта часть ядра, скажем, менеджер памяти, нарушает патент», ведь будет кошмар, верно? Это не та часть, которую можно в одночасье вынуть и заменить...

КДж: Все верно. Но если они вам скажут: «Вот патент, который вы нарушаете», вы переделаете код, обойдя патент.

LXF: Но за какое время можно написать хороший менеджер памяти?

КДж: Мы уже пару написали, а пару выкинули! [Смеется.]

LXF: Да, он сменился где-то около 2.6.10, вроде бы?..

ДА: Вполне вероятно, что люди, считающие, что текущая версия несовершенна, прямо сейчас работают над менеджером памяти и планировщиками, которые будут лучше. Если случится самое худшее, и текущая версия нарушит какие-то патенты, переключимся на другую…

КДж: Я удивлен, что они и вправду считают – есть 300 нарушенных патентов: сомневаюсь, чтобы люди, сделавшие это заявление, вообще видели исходный код и действительно имели представление, что происходит.

МФ: Они просто предположили, что при таком объеме исходного кода кто-то где-то обязательно что-то нарушил – ведь патенты носят достаточно общий характер.

2. Насущные задачи

Темпы развития в мире дистрибутивов столь быстры, что иногда, кажется, релизы разделяются буквально секундами. Какие задачи, по мнению программистов, сейчас наиболее важны, и чем сегодняшняя ситуация отличается от имевшей место пять лет назад?

Крис Джанеллони: Самая большая проблема, в моем представлении, непомерное количество вариаций оборудования. В частности, у 64-битного AMD и Intel стало так много вариантов, что практически невозможно протестировать все – да хотя бы половину!

Остается только слепо надеяться, что оно заработает. Это вызывает большие проблемы. Откровенно говоря, кажется, что всякий раз при выпуске нового релиза, день спустя выходит еще один дистрибутив, с новым ядром и новыми драйверами. Или – кто-нибудь выпускает новую материнскую плату и чипсет, не вошедшие в ваш релиз, выпущенный неделю назад. Так что, пожалуй, по-прежнему самая большая проблема – аппаратная поддержка.

Дон Армстронг: Да, поставщики оборудования, которые не сотрудничают с нами при разработке ядра, или используют предустановленные драйвера, видимо, по каким-то причинам считают, что должны оторваться от существующего слоя абстракции. Другое очень сильное изменение с изготовителями оборудования за последние пять лет – появилась такая дешевка, которая даже не может позволить себе вставить флэш-память для хранения кода на месте [в устройстве].

Поэтому поставщикам дистрибутивов приходится мучиться с оборудованием, и в большинстве случаев они не могут законно распространять [нормальные драйвера]. Или, по меньшей мере, мы не имеем доступа к исходному коду для работы с устройством, это тоже огромная проблема.

3. Debian на старом оборудовании

Debian широко известен большим числом портов на разные архитектуры. Но некоторые пользователи жалуются, что поддержка менее распространенного оборудования отвлекает силы от основных платформ, вроде x86. Прав ли Debian, что поддерживает чипы типа 68K?

Дон Армстронг: Сегодня в самом деле существует 68K, который работает примерно на 300 МГц – это встроенная система ColdFire, у нее безумная скорость. Но в том, какие пакеты под какую архитектуру собирать, решающий голос у хранителей портов. А они решили, что могут сделать это – а может, и использовать – вот и сделали.

LXF: Может быть, где-то у кого-то есть суперкластер из Atari ST!

ДА: [*Смеется*] Да, мы собираем такие для Debian – берем наши 68060-е на 75 МГц, крупные (в смысле, для Atari) SCSI-массивы, и такой объем памяти, какой только можно всадить в эти несчастные машинки. Им нужна огромная подкачка: я думаю, компиляция *libc6*, *GCC* или чего-то подобного займет добрую неделю. Ну да, все это смешно, но люди используют эту архитектуру, и действительно от разработчиков портов зависит, будут ли они ее поддерживать. Эта архитектура уже не для релиза.

LXF: Есть ли хорошие флаги CFLAGS для архитектуры 68K, чтобы их производительность сравнялась с 4-ГГц Pentium?

ДА: Не думаю, что -O способен на такой подвиг!



› Mandriva вкладывает большие усилия в дизайн Gnome и KDE, чтобы приложения одного рабочего стола гармонировали с приложениями другого.

4. Демократия в Debian

Недавно мы брали интервью у основателя Debian, Яна Мердока [Ian Murdock], и он сказал, что одно из его величайших сожалений – пришествие демократии в проект Debian: в результате тот сошел с прямого пути.

Дон Армстронг: Ну, он не совсем прав в том, что у нас демократия. Хотя мы и проводим порой голосования, в основе проекта – принцип определения места человека в нем в зависимости от его способностей. Единственный способ добиться совершения чего-то – сделать это самому. Вы можете узаконить в Debian все, что хотите, но ничего не изменится, пока люди этого не сделают.

Так что дело не в демократии. То немногое, что можно сделать демократически, это переиначивание решений по проекту, когда все говорят: «Нет, мы не собираемся делать это таким образом». И я не согласен, что демократия – это плохо. Возможно, [Мэрдок] считает, что верный путь – это доброжелательная диктатура, но у нас уже был такой опыт с Брюсом Пиренсом [Bruce Perens] – не особенно полезный.

LXF: **Люди иногда думают, что вы тратите слишком много сил на споры, а не на дела...**

ДА: У нас споры публичны, поэтому видны людям больше, чем в других компаниях вроде Red Hat. Вы не знаете о перебранках, имеющих место в Red Hat или других компаниях, потому что они идут за закрытыми дверями. Мы, в отличие от них, публично обсуждаем наши разработки: из наших списков рассылки только один является закрытым. И он не особо интересен.

Поэтому со стороны всегда кажется, что мы больше ссоримся, чем разрабатываем продукт, раз уж списки рассылки доступны общественности. У нас есть разработчики, четко нацеленные на эффективную работу, и они не участвуют в этих рассылках, им этого не надо.

Крис Джанеллони: И то же самое в Gentoo. Мы слышим точно такие же вещи, по тем же причинам: потому что все наши списки публичны, кроме одного. И единственное, что мы по нему рассылаем – это такие вещи, как уведомления с сервера, что что-то идет не так. У нас есть частный лист рассылки, но он не используется для дискуссий.

ДА: Раскрою тайну частного листа рассылок в Debian: основной его трафик – это сообщения об отпусках. Сотрудники сообщают: «На этой неделе я буду отсутствовать, и если что-то стрясется с моими пакетами, откиньте их в сторону!»

5. Gentoo: для «лихачей»?

Gentoo прошел через всплеск популярности около трех лет назад – это был его звездный час. Но постепенно его упор на тонкие настройки и производительность завоевали ему титул «дистрибутива для лихачей» (по аналогии с любителями переделок автомобилей в США, с целью форсировать скорость). Изменилось ли такое отношение?

Крис Джанеллони: Мы определенно снизили темпы роста. Но многие из тех, кто пришел к Gentoo три года назад, ушли пробовать работу с другими дистрибутивами – которые были лучшими на том этапе – и почти все перебрались на Ubuntu. Да там и остались.

Честно говоря, они-то нам, разработчикам, и доставляли головную боль, потому что требовали наибыстрейшего дистрибутива – те самые «лихачи» Gentoo. Мы потеряли много таких людей, ушедших к другим дистрибутивам.

LXF: **Бьюсь об заклад, что вы рады этому!**

КДж: Почти: сейчас много пользователей с подлинным интересом к Gentoo. Они принимают в нем больше участия, от них больше обратной связи; в целом это более позитивный опыт. Но мы, конечно, хотели бы быть в ситуации активного роста и инноваций, как было в прошлом.

LXF: **А какова роль Sabayon? Привлекает ли он больше людей к Gentoo, или, наоборот, отвлекает от общей цели проекта?**

КДж: Я думаю, ни то, ни другое. У Sabayon своя ниша; кое-кто перешел с Gentoo на Sabayon, но это же дистрибутив на базе Gentoo – он использует наши зеркала и прочее. На самом деле он отличается немногим. Зато он оттянул на себя много людей, создававших нам проблемы! И мы ему очень благодарны, и пытаемся установить лучшие отношения с некоторыми из наших «детей» [отпочковавшихся дистрибутивов]. Выясняем, что они делают иначе, чем мы, а затем обдумываем, как интегрировать какие-то вещи обратно в Gentoo.

› Крис Джанеллони работает также и с геолокацией.

› Может ли *Autopackage* решить проблему триллиона форматов пакетов (в основном несовместимых)?



6. Многовато дистрибутивов?

Сейчас имеется несколько сотен дистрибутивов Linux; одни развиваются активнее, чем другие. Но надо ли тратить столько усилий на малораспространенные дистрибутивы? Может быть, стать реалистами и как-то заставить мир Linux сконцентрироваться на, скажем, 10 Linux-дистрибутивах?

Крис Джанеллони: Действительно, есть около 10 дистрибутивов – может, и меньше – на которых основано большинство остальных: на базе Debian, Gentoo, Slackware, Red Hat, SUSE. Даже Mandrake (ныне известный как Mandriva) первоначально был основан на Red Hat. Так что главная работа сосредоточена в узкой основной области, и я, например, знаю, что многие из нас меняются заплатками, где только можно. Каждый раз, когда я ищу заплатки, первое, что я делаю – это проверяю, нет ли уже готовых у Slackware и Debian.

Так что совместной работы много. Многое из дистрибутивов-«деток» обычно создаются для заполнения особой ниши – они стремятся решить некую частную задачу лучше остальных дистрибутивов. И это у них получается, потому что разработчики родительских дистрибутивов, как правило, стремятся к большей общности.

LXF: **Вы сказали бы, что удивительный рост популярности Ubuntu во многих отношениях идет на пользу Debian?**

Дон Армстронг: В некоторой степени он был полезен, поскольку привлек внимание к дистрибутивам на базе Debian. Маркетинг и рассылка CD «ShipIt» были отлично налажены. Они, конечно, сняли с нас часть давления, создав дистрибутив для настольных компьютеров, готовый для обычного пользователя. Это позволило нам лучше делать свою работу: отступить на шаг и поддержать людей, которые используют Debian как основную ОС, людей, которые используют его на производстве, где он должен работать без сбоев все время.

Это позитивный момент. Негативный момент – отток разработчиков в Canonical. Но и тогда это были, в основном, разработчики, которые работали бы где-то еще в любом случае, так что это не всегда было большой потерей.



«Есть около 10 дистрибутивов — может, и меньше — на которых основано большинство других.»

› Майку Фрайсингеру (справа) не дают ввернуть словцо...

7. Проблема выбора?

Все мы любим свободу выбора; замечательно иметь множество программ и придирчиво отбирать те, что лучше всего соответствуют нашим потребностям. Но для базовой платформы, для системы, под которой мы фактически запускаем наши программы, действительно ли нужны и KDE, и Gnome? Это острый вопрос, но, возможно, сейчас перед нами стоит проблема: отсутствие стандартного GUI в Linux.

Дон Армстронг: А вот посмотрим, кто что использует ...

[*Упоминаются Gnome, KDE, Enlightenment, IceWM.*]

ДА: Видите, а я не пользуюсь ни одним из них. Я использую Ion3! Или WMii. [*Смеется.*] Так что выбор есть. У меня нет места для рабочего стола – я начинал с Macintosh, и сейчас я не охотник до мыши. Использую *xterm*! Вот почему это хорошо!

LXF: **Ну, мы-то как бы хакеры; а вот наши читатели стремятся ограничиться одним рабочим столом – по крайней мере, пока не заинтересуются чем-то другим. Вот они работают на KDE, и видят крутую программу, а программа-то — F-Spot. И выходит, что нужно скачать 200 МБ библиотек, чтобы установить ее... Они принимаются скачивать, и память «съедается», и приложение толком не вписывается.**

ДА: Было бы неплохо иметь обработку тем и эффективное использование виджетов, чтобы получать одинаковый вид при использовании программ *Qt* среди библиотек *Gtk* и с большей легкостью объединять их. Но я думаю, что рабочие столы настолько отличаются, и инсталляционная база пакетов, использующих разные библиотеки, настолько различается, что пока кто-то не придет и не сделает слой адаптации *libgtk-with-qt*, победителя у них не будет.

LXF: **Помню, Mandrake прилагал немало усилий, чтобы темы в Gnome и KDE выглядели одинаково – Galaxy – и это вроде как работало. Помню, я беседовал с некоторыми разработчиками, и темы были для них ужасной тратой времени, во многих отношениях.**

Крис Джанеллони: Обработка тем действительно помогла бы программам выглядеть одинаково. По-моему, особой надежды на слияние рабочих столов нет. В то же время есть много материалов на freedesktop.org, там действительно сотрудничают и стремятся привести к общему виду некоторые интерфейсы, которые в Gnome и KDE сделали по-разному.

Кроме того, некоторые другие оболочки, вроде *Xfce*, начали заниматься тем тоже. Так что другие среды могут воспользоваться результатами работы KDE и Gnome – созданием меню, к примеру.

ДА: KDE и Gnome – все-таки отдельные рабочие столы, но все больше и больше библиотек становятся совместимыми. Если сократить количество библиотек, вас не будет волновать выбор оконного менеджера – это дело вкуса. И не так уж важно, предпочитаете ли вы использовать *KPilot* или *GPilot* [инструменты Palm Pilot].

8. Форматы пакетов

Новые читатели LXF часто путаются во множестве форматов – tar-архивы, RPM, .deb и т.д. Не пришло ли время для стандартного формата пакета, типа Autopackage?

Крис Джанеллони: Я думаю, что это было бы здорово, и помоему, было бы гораздо реальнее его получить, если бы люди, выпускавшие менеджеры пакетов, делали бы их модульными, чтобы вы могли менять их интерфейс: например, добавить возможность для *Portage* [система пакетов Gentoo] читать пакетную базу Debian и узнавать, что установлено в системе. Тогда можно поставить *Portage* на Debian, или *apt* на машине с Debian, и они не будут мешать друг другу. Или способность взять и установить **.deb** – это реально работает. Я думаю, что мы зайдем дальше, чем один-формат-для-всех. Обычно в мире Linux, если что-то происходит, люди разбиваются по крайней мере на два лагеря.

Дон Армстронг: Множество людей, разрабатывая кардинально новый формат пакетов, как правило, допускают ошибки в способе управления пакетами. *RPM* и *dpkg* тоже имеют проблемы. Некоторые менеджеры пакетов переписывают существующие файлы, перемещают их неправильно, не проверяют контрольные суммы MD5 – есть масса вещей, которые трудно разрешить, но в многих случаях они уже решены.

Не обязательно создавать новый формат пакетов – используйте уже существующие: emerge, **.deb**, *RPM*, либо другие. Или сделайте их так, чтобы они легко преобразовывались с помощью *Alien*.

LXF: **RPM довольно архаичны, особенно по способу их сборки. Они созданы в те дни, когда для хранения данных использовались лентопротяжки!**

ДА: Да, это файл **cpio**. Но не что-то специфическое, вроде **.deb**; это просто архив **.ar**, с двумя tar-архивами внутри. Это довольно просто. Многое зависит от того, что делают менеджеры пакетов. Даже отвлечься от системы Debian, впрочем, вы всегда столкнетесь с проблемами: способ указания зависимостей – по имени пакета или версии. По ним не ясно, что делает пакет. У Debian хорошая политика по данному вопросу. LXF

10 лучших проектов «СДЕЛАЙ САМ»

- 3D-анаглифы из фотографий
- Защищаем Linux от детей
- Делаем загрузочный USB-брелок
- Создаем Linux медиа-сервер
- Собираем ферму для рендеринга
- Конструируем синтезатор
- Перекраиваем маршрутизатор
- Fedora на свой вкус: делаем Respin
- Создаем свои комиксы
- Широковещание web-камерой

Появление компьютеров открыло для людей уйму возможностей потратить свое время. Мы рассудили, что средний линуксоид, вероятно, пытлив – ему нравится исследовать новое и вытворять всякие фокусы со своим компьютером; разве не поэтому вы пользователь Linux?

Работа на свободной и открытой платформе позволяет людям не только создавать причудливые приложения, но и также означает, что вы свободно можете использовать их причудливыми способами. Все проекты, выбранные нами для этого обзора, доступны каждому и, как правило, не требуют других условий, чем часок-другой свободного времени и машина с Linux. Каждый проект мы свели к одной странице, чтобы вы не испортили себе глазки. Исследуйте возможности, что дарит вам Linux, и вливайтесь!

У нас прорва других проектов, которые просто не влезли в этот номер. Хотите увидеть некоторые из них?
Напишите и сообщите нам, как вы поладили с нашими проектами – можете даже присылать ваши 3D-рисунки, или ссылки на респины Fedora, или даже комиксы, если хотите!
Мы также хотим услышать, какие еще проекты вам интересны – а может, вы и сами создали проект, о котором будет интересно узнать другим? Отправьте электронное письмо на letters@linuxformat.ru



Проект 1: Создаем 3D-изображение из фото

Что нужно: Красно-голубые 3D-очки или их заменитель, фотокамера (желательно цифровая), GIMP. Время: 20 минут

Для создания достойного 3D-изображения вам потребуются две исходные фотографии, по одной на каждый глаз. Две камеры не нужны, хватит одной, если вам удастся сдвигать ее на малое расстояние и делать новый снимок. Некоторые фотомагазины продают скользящие треножники, облегчающие эту работу (или сделайте такой сами). Есть несколько важных моментов, которые следует помнить при фотографировании — обратитесь к заметкам внизу страницы.

Заполучив фотографии, остается преобразить их в два цветных изображения, а затем скомбинировать. Мы полагаем, что у вас есть популярные красно-синие очки, хотя в принципе сработают любые, если потом подогнать расцветку. А если очков нет, найдите соответствующих цветов обертки от конфет или просто кусочки целлофана, и они послужат светофильтрами.

Создание цветов

Начнем с левого изображения. В наших очках оно рассматривается через красное стекло, и для того, чтобы казаться черным или в градациях серого, ему следует быть в оттенках белого и голубого. Сделать это труднее, чем кажется: прямое раскрашивание даст вам не белый и голубой, а черный и голубой. Быстрый способ состоит в использовании слоев *GIMP*'а.

Для создания градаций серого сначала выберите Colours > Desaturate (подберите яркость). Теперь на картинке для левого глаза, видимой через красное стекло, переделаем эти градации в бело-голубые.

Продублируйте слой (Shift+Control+D), откройте диалог Layers и выберите слой Background. Его нужно заполнить голубым. Выберите Dialogs > Colours, и введите в текстовое поле **00ffff** (RGB-значение голубого).

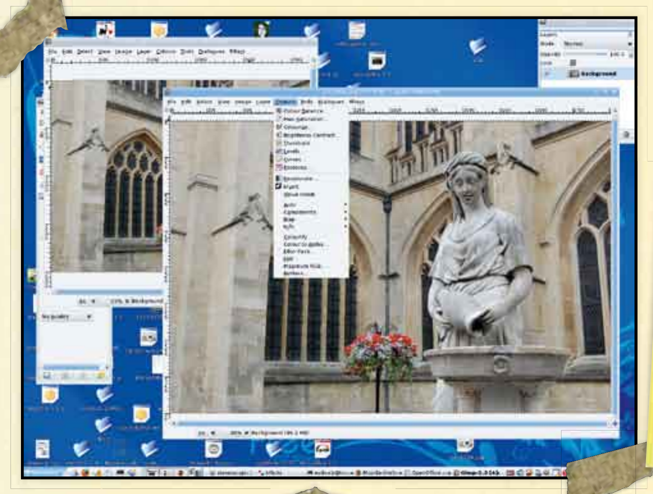
Теперь нужно залить слой Background голубым цветом, для чего нажмите Control+, (запятая). Само изображение не изменится, но вы увидите изменения в диалоге Layers. Теперь щелкните по верхнему слою диалога Layers и в списке режимов выберите Screen. Рисунок раскрасится в бело-голубые цвета. Нам потребуется плоское изображение, так что нажмите Control+M для слияния слоев.

Повторите те же действия для другого фото, на этот раз введя вместо голубого красный (**ff0000**). Итак, у нас есть два цветных рисунка, и настало время их скомбинировать.

Составление изображения

Щелкните по окну голубого рисунка и нажмите Control+A (выделить все) и Control+C (копировать). Затем перейдите в окно красного рисунка и нажмите Control+V для вставки слоя и Control+Shift+N, чтобы создать новый слой. В диалоге Layers измените режим этого нового слоя на 'Multiply'. Теперь вы видите красный и голубой рисунки одновременно. Надевайте ваши красно-синие очки и любуйтесь!

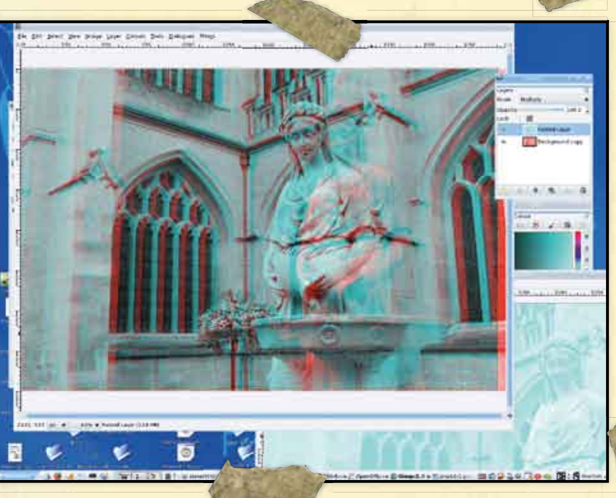
Перед уплощением изображения, вы, возможно, захотите использовать инструмент Move для подгонки положения верхнего слоя, ради наилучшего 3D-эффекта. Это поможет, например, если фотографии отличаются только по горизонтали, но в общем зависит от того, насколько удачны исходные изображения.



1 Постарайтесь выбрать сцену, имеющую хорошую пространственную глубину. 3D-эффект выходит правдоподобнее, когда объекты находятся на заметно различном расстоянии от вас.

2 Фотографируйте в ясную погоду. Используйте малую диафрагму, чтобы все было в фокусе!

3 Достали зеркалку? Используйте 50-мм объектив, более-менее совпадающий по углам обзора с человеческим глазом. При большем фокусном расстоянии картинка будет сильно искажаться на краях кадров, и достигнуть 3D-эффекта будет очень сложно.





Проект 2: Защитим ваш Linux от детей

Что нужно: Ребенок или несколько. Терпение, клавиатура без залипания. Время: Пока им не исполнится 18, а то и дольше

Маленькие дети бурлят страстью к познанию, но с таким сложным устройством, как компьютер, это чревато неприятностями. Если вы не защитите вашу машину, то пара исследовательских щелчков мышью могут угробить операционную систему. Хуже того, еще одна пара щелчков мыши в поисковой машине может открыть наименее уместную информацию из сети, так что, настраивая систему для детей, поставьте хорошее фильтрующее ПО. И вот как все это объединить.

Edubuntu «Linux для маленьких человечков» звучит лучше, чем «Моя первая ОС». В Edubuntu полным-полно обучающих приложений. www.edubuntu.org

Ktuberling Прикиньтесь, что установили это якобы для детей, и под шумок насадите физиономию вашего босса на виртуальную голову-помидор. http://games.kde.org/kde_kids.

1 Учетные записи

Для каждого ребенка, допущенного к компьютеру, обязательно создайте отдельную учетную запись: шкодливым ручонкам незачем щелкать где надо и не надо и удалять файлы с вашего рабочего стола! Для создания новых учетных записей пользователей в Ubuntu щелкните System > Administration > Users and Groups; в Mandriva и SUSE это можно сделать посредством *Control Center* и *YaST*, соответственно.

2 Укрепление

Войдите под учетной записью ребенка и сделайте рабочий стол более дружелюбным. Для новичков лучше удалить такие штуки, как апплеты сети или обновлений: они только запутают ребенка или, что опаснее, дадут ему доступ к функциям системного администрирования. Отключение виртуальных рабочих столов – тоже неплохая идея: не будут возникать вопросы типа «Куда подевались мои игры?» Если вы выбрали Gnome, щелкните правой кнопкой по объектам на панелях и прикрепите их, чтобы исключить случайное перемещение. В KDE для того же эффекта щелкните правой кнопкой по панели *Kicker* и выберите Lock Panels.

3 Выбор приложений

Теперь, имея специальную учетную запись и расчищенный рабочий стол, можно добавлять развлечения. Их выбор зависит от возраста пользователя: для самых маленьких (от 2 до 7 лет) *Gcompris* предоставляет более ста обучающих утилит и игр, построенных вокруг нашего любимого талисмана – пингвина. Вы можете найти их на DVD в разделе **Рабочий стол**. Для детей постарше добавьте *Planet Penguin Racer* и *Frozen Bubble* – отличные игры, без всякого насилия. А если вы хотите воодушевить ваше чадо на начала программирования, *Little Wizard* (http://littlewizard.sf.net) покажет, как создавать программы, объединяя иконки, а не участки кода. И всегда есть *TuxPaint*, предохраняющий вашу мягкую мебель от живописи! www.tuxpaint.org

› *GCompris* **линуксифицирует обучение пингвинами и гну.**

4 Фильтрация web

Это, безусловно, самая важная задача при создании детского рабочего стола. Когда ребенок отважится ступить в большой и грязный мир Интернета, ему нужна охрана, чтобы он/она не наткнулся на какую-нибудь гадость. (Для старших детей иногда также стоит блокировать ряд сайтов, чтобы ваши потомки не просиживали там днями напролет!) Прекрасное приложение для таких целей – *Willow* (www.digitallumber.com/willow/): прокси-сервер, фильтрующий содержимое. Его можно найти в разделе Интернет на DVD.

Скопируйте **willow-3.18.tar.gz** в ваш домашний каталог, откройте терминал и переключитесь в root через **su** или **sudo bash**. Затем введите следующее:

```
cd /var
tar xfvz ~/willow-3.18.tar.gz
```

Теперь *Willow* установлен в **/var/willow**. Перед запуском отредактируйте **/var/willow/filters/domain**, добавив в черный список сайты, какие находите нужным. Если ребенок попытается добраться до URL из этого списка из браузера или поисковой машины, он/она получит желтую страницу с предупреждением 'not accessible'. Затем введите (все еще под root):

```
/var/willow/willow.py --config=/var/willow/willow.conf
```

(Если вы получили сообщение об ошибке, касающееся модулей **exefilter**, отредактируйте **/var/willow/willow.conf** и уберите exefilter из строки filters.)

Теперь прокси запущен в фоновом режиме. Откройте браузер и настройте его на использование **127.0.0.1:8000** в качестве прокси-сервера; например, в *Firefox* зайдите в Edit > Preferences, щелкните по Connection Settings во вкладке General и выберите Manual Proxy Configuration. Введите адрес **127.0.0.1** и порт **8000**. Назначьте это для использования всеми протоколами.

Убедитесь, что все работает, просто попробовав зайти на запрещенный URL – например, playboy.com из списка по умолчанию. Теперь настройте список фильтрации доменов, добавляя туда все, что хочется [возможности Willow простираются куда дальше блокирования по имени – см. стр. 107, – прим. ред.]!

5 Резервирование

Возможно, вы не нуждаетесь в напоминаниях, но стоит повторить: последнее, что осталось сделать перед тем, как дать малышу волю – позаботиться, чтобы его домашний каталог имел резервную копию. Тогда настройки уцелеют, даже если малютка грохнет рабочий стол. Например, если Боб изловчится поломать KDE, а **/home/bob** у вас заархивирован, вы мигом извлечете домашний каталог и восстановите исходные настройки.



Проект 3: Запуск Linux с USB-брелка

Что нужно: USB-брелок на 128 MB или больше.
Время: 30 минут

Live CD прекрасно подходят для запуска ОС типа Linux, где бы вы ни были, но у них есть один вопиющий недостаток: на них нельзя записывать файлы. То есть, если вы загрузили, например, Knoppix и хотите сохранить только что созданный файл, вам потребуется иная форма съемного носителя – обычно это USB-накопитель. А не изгнать ли Live CD вообще, и не запускать ли все хозяйство сразу с накопителя USB? При помощи Feather Linux это более чем возможно! Учтите, что для данного проекта ваш ПК должен поддерживать загрузку с запоминающего устройства USB: зайдите при загрузке в установки BIOS (обычно это делается нажатием клавиш F2 или Del) и узнайте, есть ли опция для устройства USB в списке порядка загрузки.

1 Создание разделов

Для данного проекта потребуется USB-брелок на 128 МБ (или больше). Чтобы установить на него Linux и сделать его загрузочным, нужно заново создать разделы на устройстве, подобно тому, как это делается с жестким диском. При таком процессе потеряется все, что было записано на USB-брелке – так что заранее сделайте копию на другой машине!

Воткните брелок в Linux-машину; некоторые дистрибутивы тут же подмонтируют его автоматически, но мы-то хотим работать непосредственно с устройством, так что на сей раз отмонтируйте его (например, правым щелчком мыши по иконке на рабочем столе). Теперь откройте терминал и введите dmesg. Ближе к концу вывода вы увидите примерно такую строку:

```
sd 0:0:0:0: Attached scsi removable disk sda
```

Имя устройства в конце строки (здесь – **sda**) Linux присвоил нашему накопителю. Не исключено, что в вашей системе оно будет **sdb**, а то и другое какое-нибудь; тогда в последующих шагах заменяйте **sda** на то, что там выдано. Для запуска утилиты создания разделов введите

```
cfdisk /dev/sda
```

Здесь можно отвести место под установку Linux. Большинство USB-накопителей имеют один раздел, и его вы увидите в верхней части окна **cfdisk**; нажмите правую кнопку мыши и выберите опцию Delete из списка внизу, затем нажмите Enter, чтобы убрать его. (Если число разделов больше одного, удалите все.)

Теперь из нижних опций выберите New, затем Primary, и нажмите Enter. Вы создали раздел заново, как видно по списку вверху. Затем с помощью опции Maximise распространите его на весь накопитель, потом выберите Type и введите 06, чтобы была поддержка DOS FAT16. Наконец, пометьте раздел как Bootable. «Запишите» изменения на накопитель, и можете закрывать *cfdisk*.

Далее следует создать на накопителе файловую систему DOS. Введите

```
mkdosfs /dev/sda1
```

Обратите внимание на **sda1**: это означает первый раздел на устройстве **sda**. Как и выше, при необходимости поменяйте его имя на **sdb1** – точнее, на имя устройства вашего USB-брелка. Теперь следует подмонтировать устройство, так что переключайтесь на root (путем ввода *sudo bash* в Ubuntu, а в других дистрибутивах просто *su*), и монтируйте его следующим образом:

```
mkdir /usbkey
mount /dev/sda1 /usbkey
```

Теперь на нашем USB-брелке создан новый раздел и установлена файловая система, совместимая с DOS FAT16 и доступная в каталоге **/usbkey**!

2 Установка

В разделе **Distros/Feather** нашего DVD имеется последний выпуск Feather Linux – это небольшой дистрибутив на базе Knoppix, идеально подходящий для USB-брелков. Скопируйте файл **feather-0.7.4-usb.zip** в ваш домашний каталог, затем, снова в терминале, введите

```
cd /usbkey
unzip /home/username/feather-0.7.4-usb. zip
```

заменив username вашим именем. Вы увидите, что содержимое архива извлечется в **/usbkey**, куда подмонирован наш брелок. Итак, все приложения на своих местах; отмонтируйте накопитель

```
cd /
umount /usbkey
```

и теперь брелок можно извлечь. Осталось только установить загрузчик, чтобы ПК знал, как загружаться с USB-накопителя. Скачайте *Syslinux 2.11* из www.kernel.org/pub/linux/utils/boot/syslinux/Old/ и скомпилируйте его (просто распакуйте и введите *make syslinux*). Это довольно старая версия *Syslinux*, но для нашего проекта она сойдет. Затем, чтобы установить загрузчик *Syslinux* на наш USB-накопитель, введите

```
./syslinux /dev/sda1
```

3 Запуск

Теперь ваш USB-брелок готов к загрузке. Перезапустите ПК и перейдите в настройки BIOS путем нажатия F2 или Del. (на некоторых ПК кнопки могут быть другими – обратитесь к руководству пользователя). Найдите установки порядка загрузки и поменяйте их так, чтобы устройство USB оказалось первым в списке. Выйдите из настройки BIOS; теперь при загрузке ПК должен обнаруживать USB-накопитель и начинать загрузку Feather Linux!

При неудаче, попробуйте более позднюю версию *Syslinux* – например, *3.x* из менеджера пакетов вашего дистрибутива – или возьмите утилиту *mbr* из http://tinyurl.com/yt3a8e. Это небольшое приложение устанавливает новую главную загрузочную запись (MBR), так что для создания нового MBR на USB-накопителе запустите его как **install-mbr /dev/sda1**, потом заново создайте файловую систему (*mkdosfs*) и опять разархивируйте Feather Linux на USB-накопитель описанным выше способом.

› Feather Linux включает web-браузер, почтовый клиент, электронные таблицы, проигрыватель, просмотрщик PDF и много других удобных приложений.

Проект 4: Медиа-сервер для Linux

Что нужно: Маломощная Linux-машина, проводная или беспроводная локальная сеть, уйма дискового пространства. Время: 20 минут

Многие из нас накопили большие коллекции мультимедиафайлов, и хочется иметь к ним доступ с любой машины, подсоединенной к той же сети, или даже через Интернет — отсюда и популярность устройств Network Attached Storage (NAS). NAS'ы сочетают колоссальную емкость с быстротой передачи, а некоторые даже предоставляют потоковые серверы для быстрого доступа с мультимедиа-устройств или iTunes (см. раздел *Обзоры*: QNAP TS109 Pro – отличный пример). Многие из NAS'ов используют для обеспечения своей функциональности встроенную версию Linux с набором заранее настроенных открытых приложений; а стало быть, ничто вам не мешает заставить точно такие же приложения на старом Linux-компьютере выполнять то же самое.

UPnP

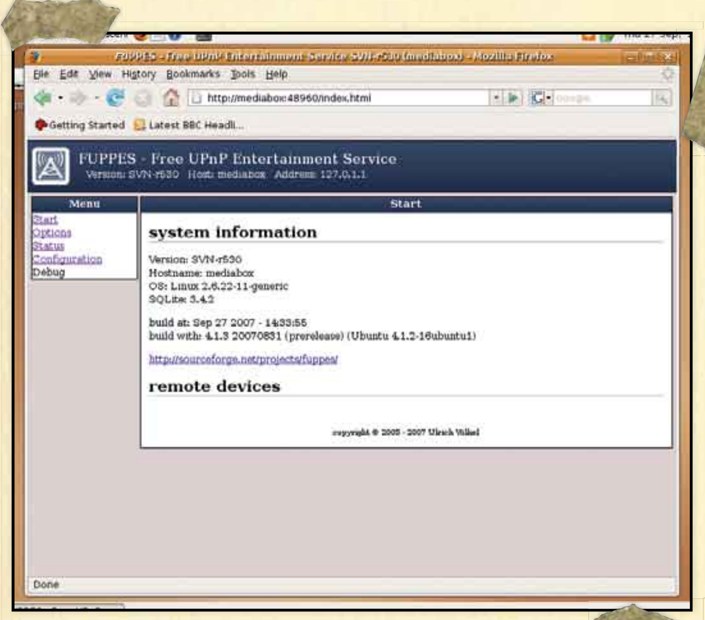
UPnP – это популярный протокол, используемый многими устройствами для передачи потока музыки, видео и фото между UpnP-совместимыми сервером и оборудованием. Например, устройства Xbox 360 от Microsoft, PS3 от Sony, SoundBridge от Roku, линейка MediaLounge от D-Link, а также N770 и N800 от Nokia могут отображать и воспроизводить медиа с сервера. Просто подключите их к сети и к телевизору – и все. Существует два свободных сервера UPnP, доступных под Linux: *Fuppes* и *Media Tomb*. *Media Tomb* – прекрасное решение, но в текущей версии не предусмотрено транскодирование. А это важно, если ваше оборудование не поддерживает непосредственно тот формат, в котором хранится мультимедиа. Playstation 3 или XBox 360 не могут проигрывать файлы Ogg Vorbis: их приходится переводить в поддерживаемый MP3. *Fuppes* умеет это делать, используя различные поддерживаемые библиотеки, и на настоящий момент является лучшим вариантом.

Если в ваш дистрибутив не включены пакеты (в большинстве случаев так и есть), тогда вам нужно скачать исходный код, скомпилировать и установить приложение самому. Сначала скачайте последнюю версию из *Subversion* по адресу http://fuppes.ulrich-voelkel.de: поскольку все меняется очень быстро, скачивание версии разработчиков – единственный способ использовать самые новые возможности. Вам также понадобится установить пакеты разработчика *sqlite3*, *lame* и *libpnp*. Используйте менеджер пакетов вашего дистрибутива, и поищите суффиксы **dev** в именах файлов пакетов. Распакуйте исходный код, который вы скачали (**tar x source.tar.gz**), перейдите в новый каталог и введите **./configure**. Вас проинформируют о недостающих библиотеках, требующих установки. В Ubuntu нам потребовалось установить *libxml2-dev, libpcre3-dev, libsqlite3-dev, uuid, taglib* и *libavformat-dev*, а также последнюю версию *ffmpeg*. Затем, для установки *Fuppes* введите **make install** под учетной записью администратора.

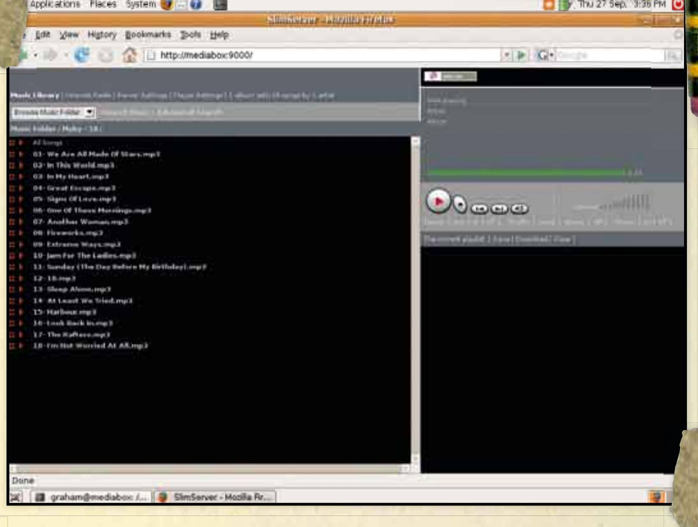
Запуск сервера теперь достигается простым выполнением команды *fuppes*. Файл конфигурации создастся в скрытом каталоге **~/.fuppes** вашего домашнего каталога, и его надо будет отредактировать, чтобы добавить расположение ваших файлов мультимедиа. После перезапуска сервера вы сможете получать доступ к вашим файлам из любого UPnP-совместимого устройства.

Музыка, iTunes и DAAP

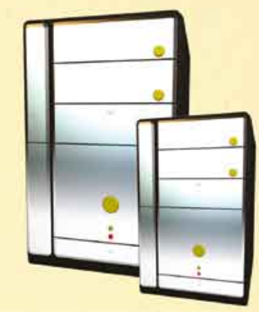
Если вы предпочитаете не фильмы/фото, а потоковое вещание, есть варианты и получше типового UPnP-сервера. Проще всего установить Slimserver, открытое ПО для устройства Squeezebox. Slimserver годится для всевозможных клиентов: чтобы проигрывать музыку, подойдет даже web-браузер! А самое приятное в нем то, что большинство дистрибутивов уже содержит готовые пакеты, и установка займет примерно 30 секунд. Однако если вы или кто-нибудь из ваших домочадцев использует iTunes, есть еще лучший вариант: Firefly Media Server. Домашняя страница проекта содержит пакеты для большинства дистрибутивов. Установка и настройка весьма несложные: просто добавьте пакет и отредактируйте одну строку в файле конфигурации, чтобы Firefly знал, где искать вашу музыкальную коллекцию. Построение базы данных займет некоторое время. При следующем запуске iTunes сервер возникнет автоматически, благодаря протоколу Bonjour, и вы сможете просматривать и проигрывать музыку с сервера так же, как и локальную. Единственным ограничением является отсутствие просмотра обложек; нельзя также синхронизировать ваши удаленные файлы с iPod'ом.



› Основанный на web интерфейс *Fuppes* следит за активностью и за сервером.



› *Slimserver* включает встроенный в браузер проигрыватель музыки, так что вы можете слушать ее везде, где есть доступ к серверу.



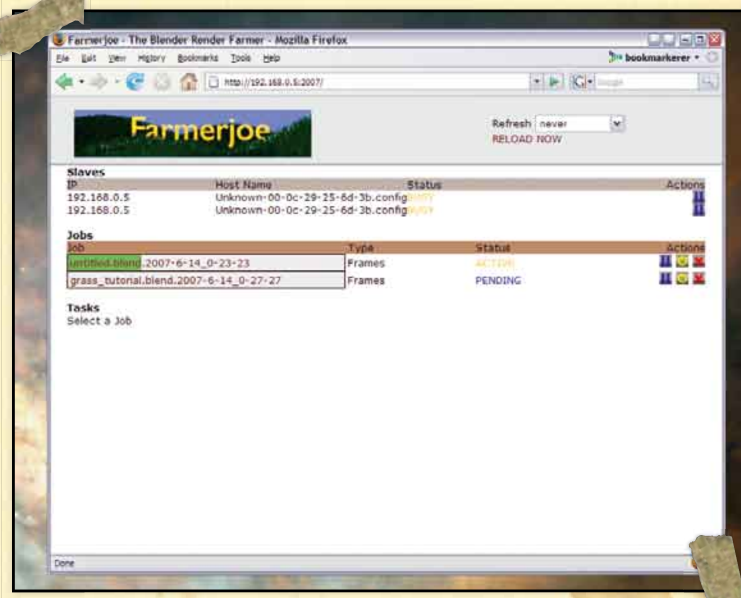
Проект 5: Ферма для рендеринга

Что нужно: Один или два свободных ПК. Локальная сеть и необходимость в дополнительных вычислительных мощностях. Время: 1 час

У всех нас по углам завалялась пара машин, которые только зря занимают место. Им, может, и слабо запустить современный рабочий стол, но как-никак они способны перемалывать миллионы операций в секунду. Хитрость состоит в том, чтобы заставить их работать параллельно, прибавляя свою мощь к вашей установке. Это называется распределенными вычислениями, и Linux для них – самая подходящая ОС: мало того, что она свободная, она еще и идеально настраивается на перераспределение задач по сети. Обычно по сети распределяют две задачи. Первая – компиляция с участием приложения по имени *distcc*: сопутствующая нагрузка на CPU делится между множеством машин. Такую компиляцию легко настроить и использовать, но тем, кто не компилирует ядро ежедневно и не работает в университетской лаборатории, проку от нее мало. Вторая задача среднему пользователю куда интереснее: это ускорение 3D-рендеринга в *Blender*, прекрасном приложении для трехмерного моделирования. При минимуме настройки машина, бездельно пылившаяся у вас под столом, резко уменьшит время, требуемое для отрисовки сцены. Чем больше компьютеров в этой компании, тем больше ускорение. Быть может, вам даже захочется использовать Live CD, чтобы призвать на подмогу машины с Windows.

1 Управление очередью

Для *Blender* есть несколько инструментов разделения задачи рендеринга по компьютерной сети. Наиболее всесторонний из них называется *DrQueue*, но настраивать и использовать его не так-то просто. Будучи хорош для профессионалов (он, например, использовался в *Пиратах Карибского моря: Сундук мертвеца*), *DrQueue* не совсем подойдет тем, кто имеет только пару свободных часов по выходным. К счастью, есть более легкий вариант. *Farmer Joe* легок в установке: просто скачайте и распакуйте архив на каждую машину, которую вы хотите использовать в коллективной отрисовке. Следующий шаг – создать общий каталог, доступный со всех машин. В большинстве дистрибутивов предусмотрен мастер Shared Directory, делающий именно это с помощью *Samba*. Убедившись, что все машины имеют доступ к общему каталогу, нужно будет на каждой из них отредактировать файл конфигурации **farmerjoe.conf**. Выберите одну машину в качестве сервера, и используйте IP-адрес этой машины для поля Master в каждом файле конфигурации. Также убедитесь, что поле linux_root указывает на общий каталог, поле linux_blender – на исполняемый файл *Blender*, а поле linux_composite – на исполняемый файл **composite**. Отредактировав все файлы конфигурации, отправляйтесь обратно на сервер и введите ./farmerjoe.linux -- master – сервер запустится. На каждом клиенте запустите./farmerjoe.linux, и они подсоединятся к серверу автоматически.

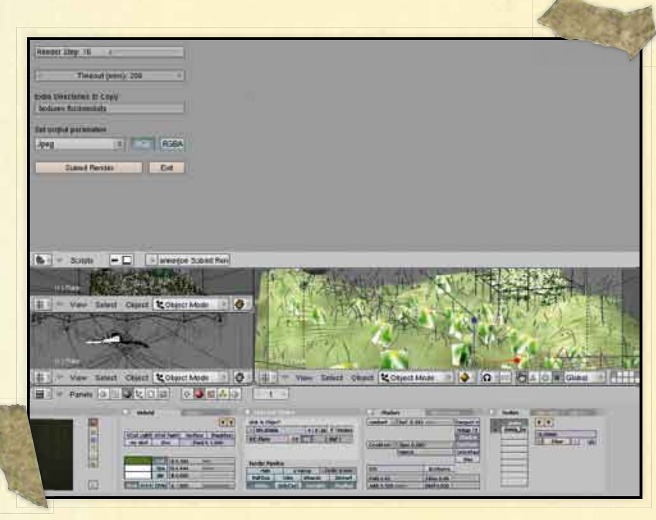


› Очередью задач можно управлять с любой машины сети, используя web-интерфейс.

2 Blend значит «смесь»

Теперь можно запускать *Blender* и загружать сцены для отрисовки. *Farmer Joe* использует скрипт для отправки задачи на рендеринг машинам фермы, и вы можете либо скопировать его из каталога *Render Joe* в каталог скриптов *Blender*, либо запустить вручную через Load New. Откроется новое маленькое окно, где нужно нажать на кнопку Submit Render. Больше ничего делать не надо. Каждый клиент примет новую задачу от сервера и примется за работу, сохраняя каждый завершенный рисунок в каталоге **jobs** общей папки. Если хотите наблюдать за ходом выполнения работы, с главной машины, можно запустить небольшой web-сервер, введя ./farmerjoe --appserver и открыв в web-браузере порт 2007 на главной машине. Вы даже можете приостанавливать и перестраивать очередь задач, если у вас появится срочная работа, требующая завершения. Вот и все: вы научились создавать собственную ферму для рендеринга.

> В настоящее время Blender Foundation работает над *Peach*, коротким продолжением *Мечты Слона*, используя *Blender* и другие открытые приложения. http://tinyurl.com/296vhx



› *Farmer Joe* использует скрипт, запускаемый из *Blender*, для добавления задач ферме рендеринга, где они ставятся в очередь на выполнение.

Проект 6: Создаем синтезатор

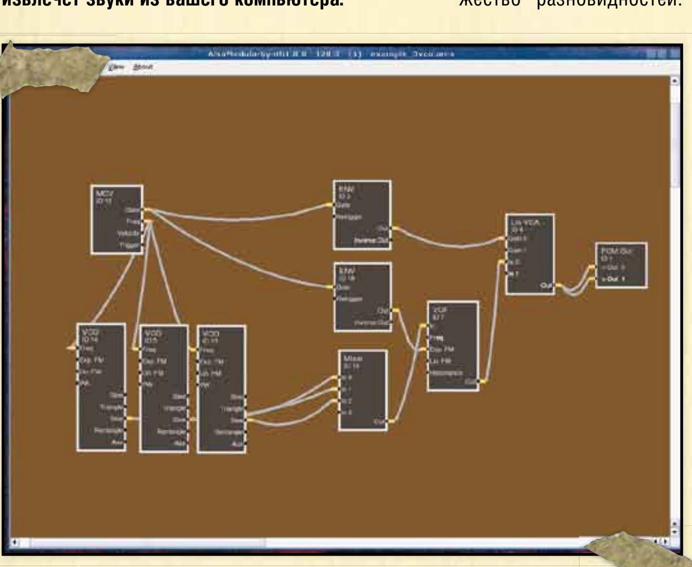
Что нужно: Способ прослушивать аудиовыход вашего компьютера и сочувствующие либо глухие соседи.
Время: 1 час и более

Синтезаторы – это здорово. Они вобрали в себя лучшие аспекты программирования, такие как сцепление модулей воедино и управление потоком данных, и используют это для создания звуков, а не приложений. И не требуется писать ни строчки кода. В Linux есть два замечательных инструмента, пригодных для создания синтезаторов. Один называется *Alsa Modular Synth (AMS)* и используется уже много лет, другой называется *Om*, он возник сравнительно недавно. Оба находятся в активной разработке и делают, в общем-то, одинаковую работу. Они используют для дизайна синтезатора графический интерфейс, а не встроенную среду разработки, а источник их вдохновения — модульные синтезаторы 1960-х и 1970-х: любое количество функциональных модулей связывается воедино внутри мешанины виртуальных кабелей. Итак, вместо целой комнаты с техникой, которая требовалась в 1976, вам потребуется только компьютер и Linux.

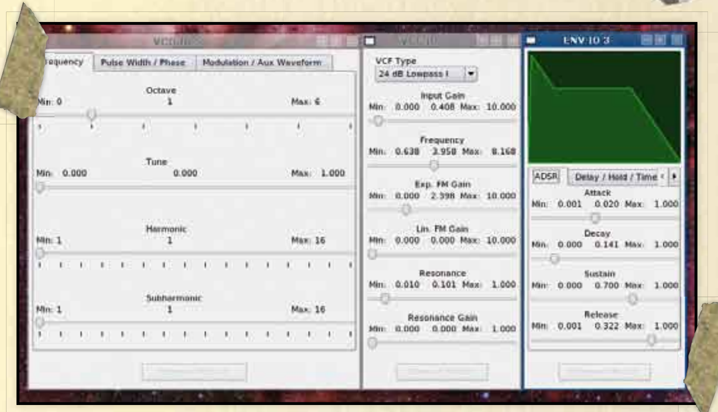
1 Типа Moog

Мы будем использовать *AMS*, так как он более стабилен и по умолчанию есть почти во всех дистрибутивах; но сделать синтезатор в *Om* ничуть не сложнее. Создадим классический трехосцилляторный синтезатор, способный генерировать старомодное звучание 1970-х. Дизайн его похож на знаменитый Minimoog, и все еще широко используется в наше время. Имеется три основных компонента. Первый – осциллятор (VCO): он генерирует начальный звук, используя предопределенную форму сигнала (как правило, синус, импульс или пилообразный). Затем этот звук проходит через вторую компоненту – фильтр (VCF), который убирает часть частотного диапазона исходной формы сигнала, добавляя глубину и тембр. Последний компонент называется модулятором и имеет множество разновидностей.

› Нет MIDI-клавиатуры? *ALSA Virtual Keyboard* извлечет звуки из вашего компьютера.



› Полная эмуляция Minimoog. Если ваш синтезатор выглядит так же, вам по плечу создание чудесных звуков.



› Для простого мягкого соло, установите примерно те же значения параметров модуля, что показаны здесь, или выберите свои!

Модуляторы изменяют уровень сигнала, добавляя огонька. Чаще всего в синтезаторах используются модуляторы амплитуды – Envelope (ENV) и низкочастотный – LFO (low frequency oscillators). Envelope меняет уровень со временем, к примеру, позволяя звуку начинаться громко и постепенно затихать. LFO постоянно добавляет модуляцию и обычно используется при добавлении к звуку эффектов вибрато или тремоло.

2 Классический модуляр

Для создания этого синтезатора, сначала запустите *AMS*. Также понадобится установить программу виртуальной клавиатуры ALSA, чтобы проигрывать ваши звуки. Внутри *AMS* вы увидите пустой коричневый экран. Правым щелчком мыши или через меню Module можно добавлять необходимые модули – нам потребуются 3xVCO, 2xENV, 1xVCF, 1xMCV, 1xVCA, 1xPCM Out и какой-нибудь микшер.

Теперь настает более сложный этап. Перетащите модуль MCV налево и, щелкнув по разу на квадратиках рядом с каждым входом и выходом, прикрепите выход Gate ко входу Gate на обоих ENV-модулях. MCV – управляющий модуль, он сигнализирует о нажатии клавиши, используемой для запуска Envelope. Затем присоедините частотный выход модуля MCV к частотным входам трех модулей VCO. Это сообщит осцилляторам, какую частоту им генерировать, то есть задаст высоту звука. Подключите пилообразные выходы всех VCO к отдельным входам микшера, чтобы суммировать звуки от всех осцилляторов, и присоедините выход микшера ко входу VCF. Выход из одного ENV направьте на вход Gain 0 на VCA (усилитель), а выход другого ENV – на вход Exp. FM на VCF, и соедините выход VCF со входом VCA. Все понятно?

Наконец, соедините выход VCA со входом модуля PCM Out – аудио отправится на звуковую карту. Вы только что собрали свой первый синтезатор, и нажатие клавиши должно издать звук. Последний шаг – установка таких параметров каждого модуля, чтобы звук получался хороший. Панель параметров возникает при щелчке по модулю правой кнопкой, а наш синтезатор так похож на Minimoog, что вы можете использовать параметры патчей, доступных для этой старой классической версии, в нашей новой программной. Взгляните на экранный снимок слева, там патч для простого мягкого соло.

Проект 7: Перекраиваем маршрутизатор

Что нужно: Ненужный маршрутизатор, с которым вы готовы распрощаться. **Время:** 2 часа

Если вы готовы рискнуть, перепрошивка маршрутизатора даст вам полный контроль над вашим оборудованием и установит на устройство ваши собственные утилиты, превратив его из пассивного кирпича в маленький компьютер с Linux. Получив над ним контроль, можете установить почти любой сервер Linux, какой пожелаете, включая платформу телефонии *Asterisk*, DHCP, DNS, VPN и многое другое. В некоторых маршрутизаторах вы даже получите доступ к заблокированному оборудованию или сможете разогнать устройство и выжать из машины дополнительную скорость.

Зачем ломать идеально работающий маршрутизатор на гарантийном сроке, рисковать стабильностью и поддержкой? Да, собственно, незачем! Дождитесь, пока ваш старый маршрутизатор пойдет на выброс. Нужен также солидный опыт работы с Linux: если что-то пойдет не так, подсоединить клавиатуру и мышь и найти неисправность с экрана не удастся. Недокументированные действия с маршрутизатором ведут к риску потери гарантии, а также есть вероятность полного выхода вашего оборудования из строя.

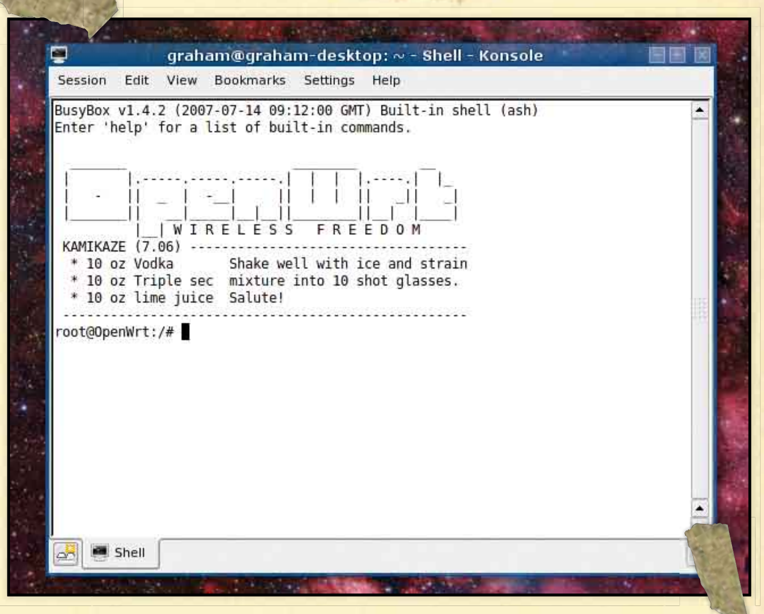
Открытая прошивка

Все это возможно благодаря *OpenWRT*, проекту, который стартовал как замена для Linksys WRT54G, но с тех пор дорос до охвата десятков других маршрутизаторов. И вот первый шаг: убедитесь, что ваш маршрутизатор поддерживается. Просмотрите длинный список поддерживаемых устройств на **http://toh.openwrt.org**. Поддерживаются многие устройства от Asus, Belkin, Buffalo, D-Link и Netgear, хотя львиную долю устройств все еще занимает Linksys. Ссылка в таблице приводит к инструкциям для вашего устройства, а также к специальным программным прошивкам, которые заменят те, что стоят у маршрутизатора по умолчанию. Если вашего устройства нет в списке «поддерживаемых», то использовать его нельзя, хотя там имеется несколько, помеченных «в разработке», с которыми может все получиться. Как правило, трюк с заменой ваших программных прошивок состоит в перехвате процесса загрузки маршрутизатора, но для разных маршрутизаторов он разный, и вам следует свериться с таблицей совместимости и следовать инструкциям именно для вашего оборудования. Но недавно были созданы двоичные пакеты, имитирующие официальные обновления прошивок для всех распространенных маршрутизаторов – то есть можно установить *OpenWRT*, не пользуясь ничем, кроме кнопки Обновить web-интерфейса вашего маршрутизатора. Несколькими мгновениями спустя, после перезагрузки маршрутизатора, у вас будет установлен *OpenWRT*, малой кровью. Только вы не сможете разговаривать: по умолчанию *OpenWRT* пока не включает пользовательский интерфейс, и единственный интерфейс к вашей новообретенной свободе – почтенный telnet. Первым делом подсоединитесь к маршрутизатору и смените пароль администратора. Просто введите telnet, затем – IP-адрес маршрутизатора (он зависит от конфигурации сети, однако если вы совершенно не в курсе, попробуйте 192.168.0.1). Вы автоматически подсоединитесь, так как защиты паролем пока нет. Переделайте это, введя passwd, а потом – новый пароль администратора. Теперь после перезагрузки маршрутизатора доступа по telnet уже не будет. Вместо него можно использовать заданный пароль для доступа через SSH – куда более безопасный вариант. Это Linux, и ваши обычные навыки будут творить чудеса и здесь. Введите help, чтобы увидеть список распространенных команд, но без ipkg вы далеко не уйдете: эта команда устанавливает на маршрутизатор дополнительные пакеты. Введите ipkg update, чтобы скачать последний список доступных пакетов, затем выведите этот список по ipkg list. Первое, что вы, наверное, захотите установить – это *X-Wrt*, управляющую web-консоль. К вящей путанице, доступны и другие прошивки, по умолчанию включающие *X-Wrt*, но web-интерфейс можно установить вручную, набрав

```
ipkg install http://ftp.berlios.de/pub/xwrt/webif_latest.ipk -force-reinstall
```

Ваш маршрутизатор перезагрузится, и вы сможете управлять высвобожденным оборудованием, открыв web-браузер и указав IP-адрес маршрутизатора. Если вы ранее пользовались утилитой Webmin, то будете чувствовать себя, как дома.

› Основной интерфейс *OpenWRT*, по крайней мере, поначалу – это telnet. Вот что вы увидите при первом соединении.



Проект 8: Создайте свою Fedora

Что нужно: Дистрибутив Fedora 7, Linux с привилегиями root'а. Время: 2 часа

Существует много причин стремления создать свою версию дистрибутива Fedora, и это намного проще, чем вы думаете. Много лет назад спецы Red Hat и сообщество Fedora создали поразительное компактное скриптовое приложение *Revisor*. Идея его состояла в том, чтобы запросто пересобирать Fedora, добавлять собственное содержание или даже заранее выбирать приложения для включения. На заре юности *Revisor* имел причуды, но сейчас он гораздо лучше!

1 Начало

Во-первых, скачайте *Revisor* с http://revisor.fedoraunity.org. Правда, имеется несколько зависимостей, которые тоже потребуется установить, но не особо сложных. Скорее всего, нужны будут только *livecd-tools* и, быть может, дополнительные пакеты Python'а. В идеале хорошо бы у вас уже работала Fedora 7, и не обойтись без привилегий суперпользователя.

```
yum install revisor
```

Объем загрузки не превышает 15 МБ. Теперь можно запустить *Revisor*, просто введя команду в окне терминала. Вас попросят ввести пароль root'a, и после этого все заработает.

Заглавная страница предложит вам варианты отображения продвинутых опций, которые следует учесть, если вы хотите делать что-то кроме CD/DVD. Щелчок по Next приведет вас в настройку Media Type (типа носителя). Здесь важно решить, хотим ли мы устанавливать систему с нашего дистрибутива или просто загружаться с него. Вы также можете сделать виртуальный установочный набор, но, возможно, это уже слишком круто. Для нашего проекта мы просто сделаем Live CD с PDF-браузером и кое-какими старыми номерами журнала, то есть нам надо выбрать Live media.

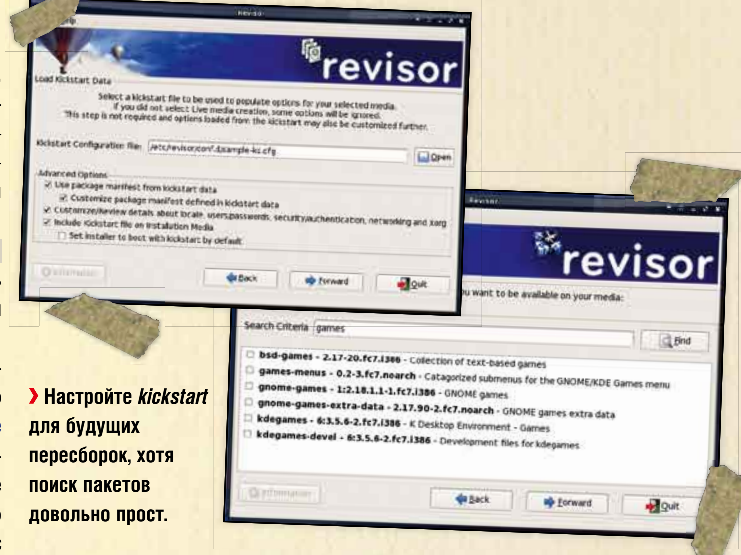
Следующая страница попросит указать тип носителя. Мы выберем Optical.

2 Repo man

Следующий раздел – настройка ваших репозиториев. Выпадающий список Section предлагает выбрать базовый код для вашего дистрибутива. Версия разработчиков – это *Rawhide*, а *fc7* в настоящее время – релиз [к тому моменту, когда вы будете читать эти строки, релизом будет fc8, – прим. ред.]. Выбрав конфигурацию, вы увидите список репозиториев. При желании его можно пополнить (например, добавить *livna*). Если вы хотите добавить в дистрибутив собственные файлы, следует создать для них пакет и настроить локальный репозиторий для их обслуживания. Смотрите на желтую подсказку вверху!

› Даешь эффективность 3D-рабочего стола! В разработке *Revisor Beryl*.

Хотите добавить свои файлы? Установите пакет *createrepo* и просто запустите его в каталоге, содержащем эти файлы, для генерирования заголовков. Затем вам нужно запустить FTP- или HTTP-сервер, чтобы позволить *Revisor* выбрать пакеты.



› Настройте *kickstart* для будущих пересборок, хотя поиск пакетов довольно прост.

Затем выберите себе пакеты, почти тем же способом, как при первой установке Fedora. Все организовано по группам, и зависимости разрешаются автоматически. Чтобы вы могли прикинуть, во что это выльется, скажем, что минимально возможный размер – 480 МБ, но учтите: сюда не включено даже окружение рабочего стола!

Следующая стадия – самая длинная: этап сборки, когда *Revisor* получает необходимые пакеты. Затем проходит процесс завершения установки (он может тянуться еще дольше, так что наберитесь терпения). Если вы предпочли Live-дистрибутив, то вдобавок понадобится «установить» пакеты на сжатую файловую систему, откуда они будут запускаться, а также настроить загрузчик. Конечным результатом будет ISO-образ, который можно прожечь на диск и наслаждаться, наслаждаться!

Конечно, если вы хотите чего-то посолиднее, потребуется повозиться с пакетами и файлами *kickstart*. Файл *kickstart* – это просто файл, который сохраняется в конце установки Fedora, с журналом всех сделанных вами выборов. Его можно использовать для автоматизации будущих установок или для определения набора пакетов, предназначенных для создания новой личной версии Fedora. С виду кажется, что потребуются большие усилия, но вы всегда можете загрузиться на виртуальной машине, которую вы так и так используете для тестирования!

Проект 9: Создаем комиксы в Inkscape и Gimp

Что нужно: Дар художника, способности рассказчика. Время: долго

Мир требует все больше комиксов. Правда-правда. Можете включиться в этот процесс — это проще, чем вы думаете. Перед стартом следует уяснить несколько моментов:

1 Есть масса способов создания рисунков и окончательного документа; показан только один. Хотите делать по-другому? Вперед!

2 Прочтя этот проект, вы не станете лучше рисовать. Улучшить рисование можно только рисуя.

3 Вообще-то схема применима и без рисунков – визуализируйте 3D-сцены или пользуйтесь фотографиями, подобно любовным фотоисториям Jackie или долгожданному 1985 Марка Миллара [Mark Millar].

Положим, что вы все-таки собираетесь рисовать комикс. Вариантов ввести рисунки в компьютер существует несколько. Можно, конечно, прямо там их и нарисовать – наличие графического планшета здорово упростит задачу. А если вы сделали рисунки на бумаге, отсканируйте или сфотографируйте их. В зависимости от стиля рисунка, даже дешевая цифровая камера может дать приличное качество. И сканированные, и сфотографированные рисунки, скорее всего, сперва потребуют небольшой чистки, что, возможно, лучше всего делает *GIMP*. Но мы забегаем вперед.

Размер имеет значение

Определитесь с размером комикса и создайте документ в *Inkscape* того же размера. Стандартный размер комикса в США – 6 5/8''x 10 1/4''. В Британии стандарта нет (знаменитый *«2000 н.э.»* выходил во всех размерах); если хотите, используйте A4. Помните про поля страниц; прикиньте, сколько места потеряется в зависимости от переплета – скрепками или клеем.

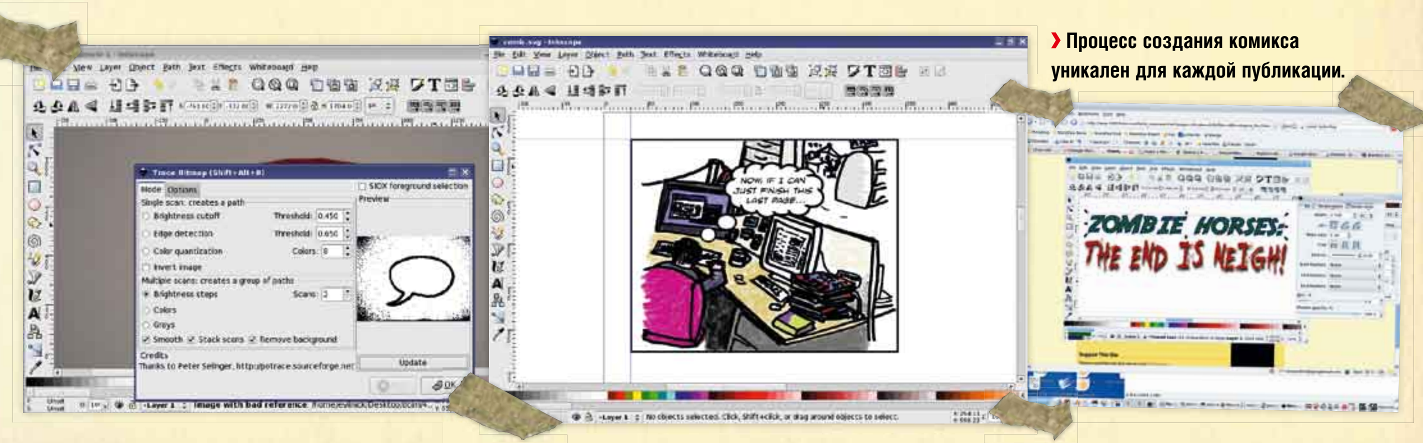
Далее следует сделать набросок вашей страницы и примерно представить ее раскладку. Размер и форму кадров полезно знать заранее, чтобы сразу рисовать правильно. Кадры рисуйте по отдельности. Для лучшего результата, рисуйте как профи – сначала наметьте мягким карандашом, затем обведите линии толстой кистью или пером. Большинство художников делает рисунки в 120 % от окончательного размера, а то и больше, а вы можете работать при 150 % и использовать реально толстое перо для контуров и более тонкое для деталей. Не забудьте оставить место для пузырей изо рта, которые с текстом!

Загоните рисунки в компьютер через сканер или камеру и обработайте их *GIMP*'ом. Повысьте контрастность и яркость: серые пятна исчезнут, останутся только темные контуры. Переключитесь в режим индексированных цветов, потом в монохромный, затем обратно в градации серого, и сохраните продукт. Теперь рисунки можно импортировать в *Inkscape* и разместить по кадрам. Если вы в основном рисовали жирными линиями, можно получить изящный эффект очистки, трассировав растр и создав из него контуры. Это сгладит некоторые области рисунка и приведет к более гладкому результату, но может исказить ваш стиль.

> **Добавление цвета**
>
> Вам хочется цвета – прелестно! Существует чудный способ его добавить. Экспортируйте все страницы из *Inkscape* как растровые рисунки. Потом загрузите их в *Gimp* и раскрасьте по велению вашего сердца. Контуры можно использовать для плоской заливки, градиентов или чего-нибудь еще, а можно раскрашивать области. Главное, что не нужно беспокоиться о выходе за линии, как вы сейчас увидите. Снова сохранитесь в виде растра и переключитесь в *Inkscape*. Прежде чем делать что-то дальше, выберите Layers > Add New Layer и создайте новый слой. Теперь импортируйте раскрашенный растр. Подгоните его так, чтобы он идеально совпал со страницей, чего легко добиться, введя 0 в поля смещения по x и y в панели инструментов. Теперь нажмите Shift+PgDown, чтобы поместить этот слой ниже всех, и рисованный слой закроет все участки, где вы заехали на линии. Ура!

Чаво сказал?

Думаете, в *Inkscape* создавать пузыри с текстом просто? Может, вы и правы, но они не будут выглядеть как полагается. В зависимости от стиля вашего комикса, может быть лучше нарисовать несколько вариантов, отсканировать их и преобразовать в SVG-объекты *Inkscape*. Хороший совет – нарисовать сначала диалог: легче оформить пузырь вокруг нормального, хорошо выглядящего куска текста, чем пытаться втиснуть слова в нарисованную фигуру. Сами буквы, конечно, можно выполнять от руки, но в наши дни профессионалы используют шрифты (обычно собственные, выполненные своим почерком). К счастью, множество шрифтов для комиксов доступно как Freeware или вообще Public Domain. Подберите шрифты комиксов на **www.1001fonts.com** или **www.blambot.com/fonts.shtml**, скопируйте их (под root) в вашу папку **/usr/share/fonts** и перезапустите *Inkscape*, чтобы их можно было выбирать. Для надписей больше подходят шрифты, имеющие несколько размеров, они дают возможность выделения увеличением. Нужно ли говорить, что легче создавать их на отдельном слое, заблокировав остальные, чтобы предотвратить смешение с расположенным ниже рисунком.



› **Процесс создания комикса уникален для каждой публикации.**

Проект 10: Широковещание web-камерой

Что нужно: Web-камера, совместимая с Linux, постоянное интернет-соединение, Asterisk необязателен. Время: 15 минут

Все мы уже купили web-камеры пару лет назад. Тогда это казалось отличной идеей. Мы полагали, что сможем на халяву болтать с друзьями и родными, видя их радостные лица, созерцающие нас. Увы, даже когда это удавалось, первые полчаса тратились на устранение по телефону нестыковок с их web-камерой, что убивало всю идею. Так что в основном web-камеры валялись забытые по коробкам, в компании прочего хлама. Но сейчас их время пришло. Используйте вашу web-камеру для широковещания маленького окна в вашу жизнь на остальной мир (или на локальную сеть, если вы скромны).

1 Web-камера

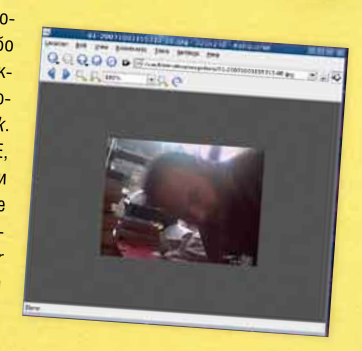
Это покажется очевидным, но самое главное в этом проекте – убедиться, что ваша web-камера должным образом поддерживается. Linux проявляет разборчивость в этом вопросе. Старая Phillips PCVC675K не создала нам проблем, но мы знаем более новые устройства, которые не работают. Простейший способ протестировать вашу камеру – подключить ее к машине и запустить приложение, показывающее видео, например, *Ekiga*. Если это не работает, ищите в базе данных поддержки вашего дистрибутива любые намеки на то, как приставить устройство к делу. Если работает, то следующим шагом будет установка программы, делающей снимки и отсылающей их на сайт. Лучшей небольшой утилитой, которую мы нашли, оказалась *webcamd*. Это – простой демон, выполняющийся в фоновом режиме: он захватывает рисунок с web-камеры через заданные промежутки времени и копирует полученный файл на локальную web-страницу или FTP-сервер (два наиболее распространенных способа хостинга потока с web-камеры). *Webcamd* надо установить через менеджер пакетов вашего дистрибутива, а если вы хотите запустить еще и локальный web-сервер, потребуется установить еще и *Apache*.

2 Дни демона

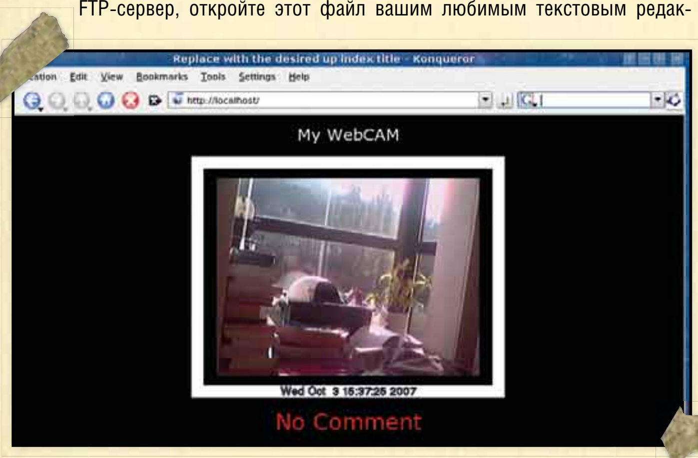
После установки, выберите пользователя, под которым вы хотите запускать демона *webcamd*. Если вы используете *Apache*, убедитесь, что этот пользователь имеет право на запись в каталог web-содержимого (обычно **/var/www**). Переключитесь на этого пользователя и введите webcamd в командной строке. Создастся файл конфигурации **~/.webcamd/webcamd.conf**. Если вы хотите загружать изображения на FTP-сервер, откройте этот файл вашим любимым текстовым редактором, измените enable_ftp = no на enable_ftp = yes и убедитесь, что настройки сервера соответствуют IP-адресу, имени пользователя и паролю, требуемым вашим FTP-сервером. Может быть, вы захотите изменить в этом файле и другие параметры, например, скорость обновления и разрешение изображения. Наконец, закончив все исправления, просто введите webcamd start для запуска демона в фоновом режиме. Если у вас запущен *Apache*, вы сможете указать web-браузеру ваш IP-адрес, и тогда вы увидите страницу с внедренным в нее изображением с web-камеры. Если вы используете FTP-сервер, то изображение загрузится на сервер. Все, что вам останется сделать – это изменить встроенную web-страницу на ваше собственное усмотрение. LXF

Почувствовать движение

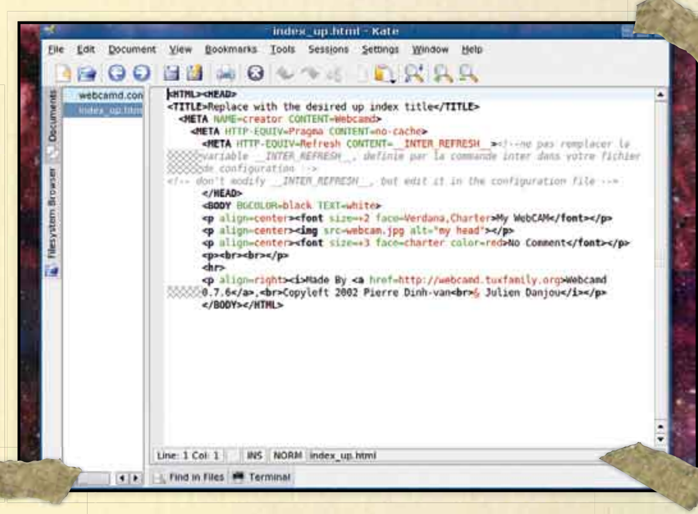
Если вам удастся получить работающую web-камеру, примените ее практически, употребив как сенсор движения. Такой используется в *ZoneMinder*, домашней системе безопасности, которая опрашивает web-камеры на предмет засеченного движения. Что-нибудь обнаружив, она может либо подать тревогу, либо послать электронное письмо, либо даже позвонить, если у вас используется *Asterisk*. *ZoneMinder* также увязан с LinuxMCE, домашним медиацентром (мы писали про него в LXF98), так что вы можете превратить ваш ТВ в Linux-эквивалент HAL9000. Но если *ZoneMinder* слишком роскошен для нужд вашей безопасности, обратите внимание на небольшое приложение командной строки под названием *motion*. Оно включено в большинство дистрибутивов, а зависимостей у него нет. *Motion* великолепно своей простотой: просто наберите motion в командной строке. При обнаружении какого-либо движения в ваш компьютер записывается снимок (по умолчанию, в **/var/lib/motion/snapshots/**). Очень просто.



› Используйте команду *motion* для поимки воров, прокравшихся к вам ночью за вашей коллекцией фарфора.



› Поделитесь видом из окна вашего офиса с остальным миром, настроив широковещание с рабочего места.



› Можно настроить вывод демона web-камеры, отредактировав простой HTML-файл, расположенный в каталоге конфигурации.





10 лучших расширений для WordPress

Правду ли говорят, что лучшее – враг хорошего? Чтобы дать ответ на этот вопрос, **Валентин Развозжаев** решил рассмотреть десятку лучших расширений к лучшему блог-движку.

В апреле 2007 года Яндекс опубликовал отчет о состоянии блогосферы российского Интернета. По его данным, общее число русскоязычных блогов достигло двух миллионов (всего в мире 76 миллионов блоггеров), каждую секунду в сетевых дневниках появляется в среднем пять новых записей, а среднестатистический блоггер – это девушка 21 года, которая живет в Москве и учится в ВУЗе. Любопытно, что интересы среднестатистической девушки-блоггера за последние несколько лет ничуть не изменились: это музыка, кино, психология, книги и секс.

Хотя большинство блоггеров размещают свой дневник на одном из стандартных блог-хостингов (чаще всего LiveJournal.com и LiveInternet.ru), также растет и количество «автономных» блогов – к марту этого года оно достигло девяти тысяч. Среди движков, которыми пользуются автономные блоггеры, безоговорочно лидирует *Wordpress* (его доля составляет 71%, в то время как доля каждого из остальных движков – менее процента).

Популярность *Wordpress* вполне закономерна: движок очень просто устанавливается и настраивается, поддерживает web-стандарты XHTML и CSS, позволяет выбрать тему оформления, поддерживает публикацию записей с помощью сторонних программ и сервисов. Для него разработано огромное количество расширений, способных удовлетворить все мыслимые и немыслимые потребности пользователей.

Загрузить последнюю версию движка можно как с официального сайта разработчиков (http://wordpress.org/), так и с ряда других сайтов (например, http://maxsite.org/). В последнем случае вы получите полностью локализованный движок с набором предустановленных расширений, что очень удобно.

Установка *Wordpress* очень проста. После загрузки архива распакуйте его в локальный каталог. Затем задайте настройки соединения с базой данных в файле **wp-config-sample.php** и переименуйте его в **wp-config.php**. После этого загрузите все файлы в корневой или любой другой каталог web-сервера. Затем откройте в браузере инсталляционный скрипт http://адрес_блога/wp-admin/install.php и следуйте указаниям мастера.

С помощью административной панели http://адрес_блога/wp-admin/ можно задавать необходимые настройки движка, добавлять, редактировать и удалять записи, модерировать комментарии, управлять пользователями и т.д. На вкладке Плагины можно включать и отключать требуемые расширения, а также править их исходный код.

Для установки расширения нужно скопировать исходные файлы в каталог http://адрес_блога/wp-content/plugins/ и затем активизировать его на вкладке Плагины. Иногда необходимо также внести изменения в файлы темы (каталог http://адрес_блога/wp-content/themes/текущая_тема/). Более подробная информация по установке расширения обычно содержится в файле **readme.txt** в архиве или на web-странице проекта.

Наиболее полный список расширений *Wordpress* приведен на странице http://codex.wordpress.org/Plugins. Они разбиты по категориям: администрирование, реклама, управление комментариями, статистика, оптимизация блога в поисковых системах и т.д. В этой статье мы рассмотрим десятку наиболее интересных расширений, охватывающих почти все категории. И начнем с управления комментариями.

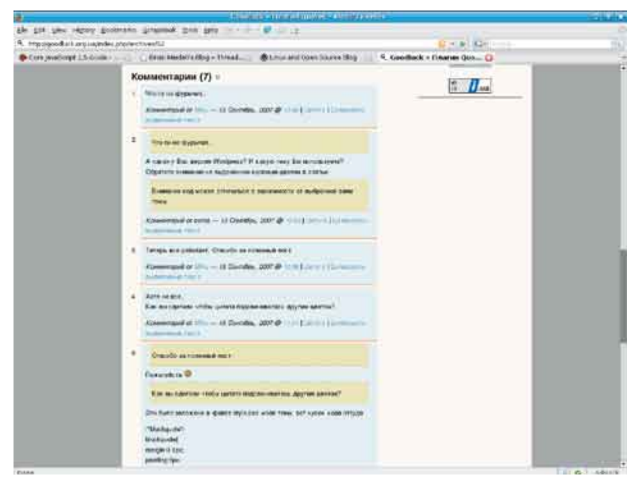
Комментарии

В оформлении комментариев в *Wordpress* есть пара недостатков, которые бросаются в глаза. Первый – при добавлении нового комментария нельзя цитировать предыдущие. Второй – комментарии располагаются не иерархически (как, например, в *LiveJournal*), а один под другим. К счастью, и то, и другое поправимо.

1 Одним из решений первой проблемы служит расширение *Quoter* (http://www.damagedgoods.it/wp-plugins/quoter/). Оно может использоваться как для цитирования текста комментария, так и произвольного текста на странице.

Настройки расширения (отображаемый текст, форматы даты и времени) можно задать на странице Настройки – Quote административной панели. Внешний вид формы добавления комментариев можно изменить путем редактирования файла http://адрес_блога/wp-content/themes/текущая_тема/comment.php.

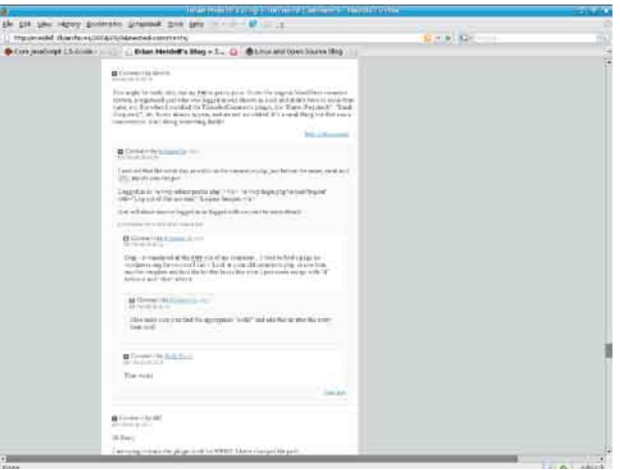
После внесения изменений в этот файл и активизации расширения в каждом комментарии появляется ссылка, с помощью которой можно процитировать его в ответе. Последняя на момент написания статьи версия *Quoter* позволяет цитировать комментарий только целиком.



› Рис. 1. Расширение *Quote*.

2 Для решения второй проблемы подойдет расширение *Brian's Threaded Comments* (http://meidell.dk/archives/2004/09/04/nested-comments/), с помощью которого можно организовать иерархию комментариев: группа связанных друг с другом комментариев будет отображаться в виде «лесенки».

Подробные инструкции по установке расширения можно найти в файле **install.txt** в архиве программы.



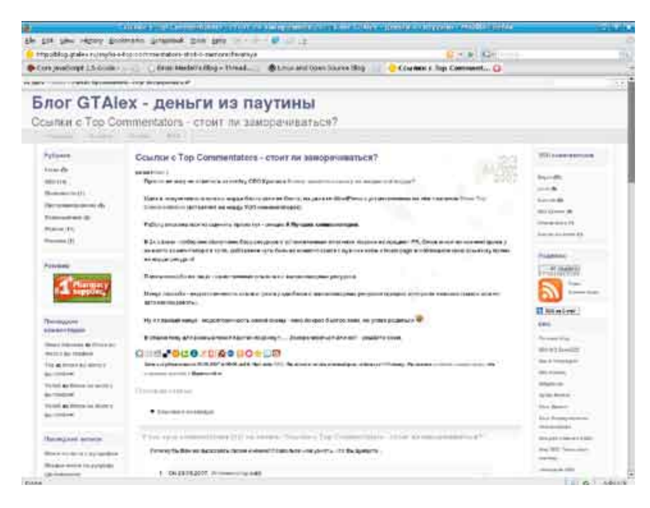
› Рис. 2. Расширение *Brian's Threaded Comments*.

3 С помощью расширения *Show Top Commentators* (http://www.pfadvice.com/wordpress-plugins/show-top-commentators/) можно сформировать список посетителей сайта, оставивших наибольшее количество комментариев, и вывести его на всеобщее обозрение прямо в блоге.

Нужно отметить, что вместе с приятными ощущениями каждый из постоянных посетителей вашего сайта может получить дополнительную ссылку на свой блог. Для этого ему нужно лишь указать адрес своего сайта при добавлении комментария. Дополнительная ссылка на сайт, размещенная на посещаемом ресурсе, может повысить его Page Rank в Google и благоприятно сказаться на посещаемости.

Настройки расширения *Show Top Commentators* можно задать с помощью редактора плагинов (Плагины > Редактор плагинов) или путем редактирования файла **show_top_commentators.php** вручную. Основные настройки включают в себя:

- задание периода сбора статистики, по истечении которого она обнуляется (час, день, месяц, год);
- максимальное количество пользователей в списке;
- исключение из статистики отдельных пользователей (например, администратора);
- включение или отключение показа числа комментариев для каждого пользователя.



› Рис. 3. Расширение *Show Top Commentators*.

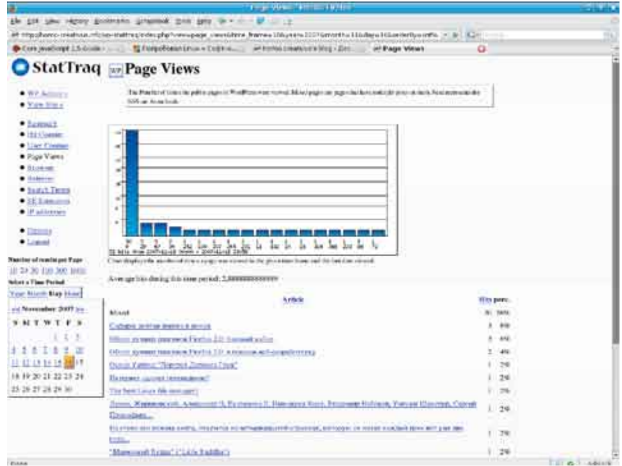
Статистика

Статистика знает все. Информация о самых читаемых и самых комментируемых записях, используемых посетителями браузерах, реферерах блога и его положении в поисковых системах может быть интересна не только автору дневника, но и его читателям. Между тем, стандартные блог-хостинги не предоставляют пользователям практически никакой статистики посещаемости ресурса. К счастью, в случае с *Wordpress* все обстоит по-другому. И сейчас мы в этом убедимся.

4 Начнем с расширения, которым можно было бы и закончить. С помощью *Stattraq* (http://thefunzone.awardspace.com/Wordpress/?page_id=63) можно получить исчерпывающую статистику использования блога, в которую входят следующие показатели.

Информация о хитах. Под количеством хитов понимается число обращений пользователей к страницам сайта. Для каждого обращения приводится информация о дате и времени обращения, IP-адресе посетителя, реферере (ссылке, которая привела посетителя на ваш сайт) и информация об используемом браузере.

Число пользовательских сеансов. Как и в предыдущем случае, для каждого сеанса приводится информация о дате и времени доступа, IP-адресе и браузере пользователя, а также о просмотренных страницах.



› Рис. 4. Расширение *Stattraq*.

Просмотренные страницы. Показатель представляет список наиболее популярных страниц сайта с указанием числа просмотров каждой страницы. Страницы сайта, на которых размещено более одной статьи (например, титульная страница или страницы рубрик), объединяются в один пункт списка под названием Mixed. Обычно этот пункт занимает верхнюю строчку в списке.

Браузеры. Список наиболее популярных браузеров, для каждого из которых указывается общее число обращений.

Рефереры. Реферер блога – это сайт, с которого посетитель перешел к вам. Выводится список рефереров блога, для каждого из которых указывается страница вашего сайта, на которую перешел пользователь.

Ключевые слова. Список ключевых фраз, результаты поиска по которым в поисковых системах Google, Yahoo и MSN содержали ссылки на страницы вашего сайта, по которым пользователи и пришли к вам. Этот показатель формируется на основе предыдущего. Ключевые слова для поисковой системы Yandex пока не попадают в этот список, впрочем, ничто не мешает вам найти их самостоятельно в списке рефереров.

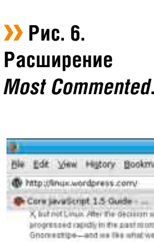
Положение в поисковых системах. Число проиндексированных страниц блога в каждой из поисковых систем: Google, Yahoo и MSN и общее число страниц.

IP-адреса пользователей. Список IP-адресов с указанием даты и времени доступа, используемого браузера и информации о просмотренных страницах для каждого IP-адреса.

Статистику по каждому из этих показателей можно формировать для конкретного временного интервала (час, день, месяц, год). Например, можно получить информацию о просмотренных страницах, IP-адресах пользователей и рефферах сайта за прошедшие сутки, за определенный месяц или год. Для большинства показателей производится автоматическое построение графиков.

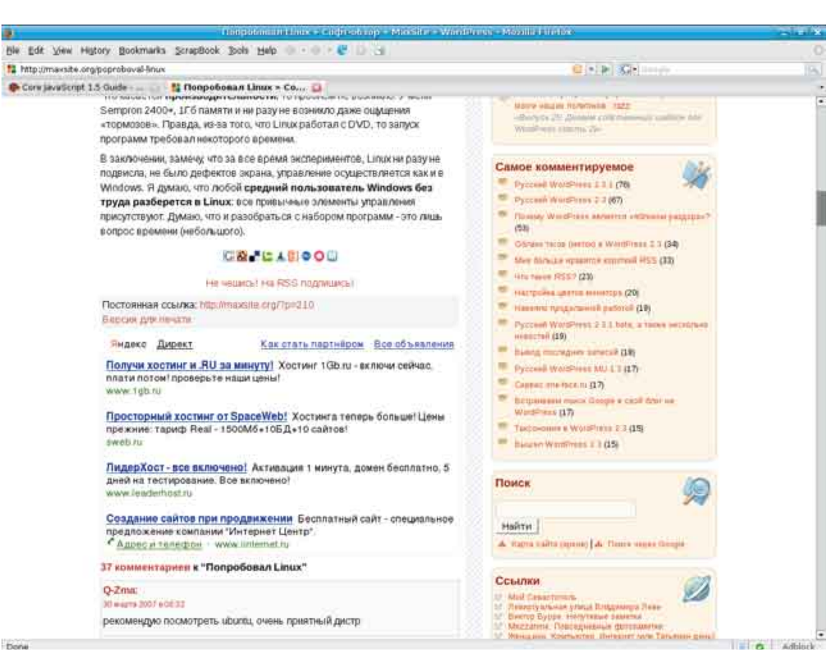
Настройки расширения позволяют разрешить или запретить доступ к странице со статистикой незарегистрированным пользователям, а также исключить из формирования статистики визиты роботов поисковых систем и агрегаторов RSS.

После установки расширения для доступа к странице со статистикой используйте адрес http://адрес_блога/wp_Stattraq/. Как правило, статистика формируется за несколько часов.





› Рис. 5. Расширение *MostWanted*.



›› Рис. 6. Расширение *Most Commented*.

5 На основе расширения *Stattraq* были разработаны многие другие, среди них – *MostWanted* (http://boakes.org/most-wanted), с помощью которого можно получить список самых популярных статей блога. Для формирования этого списка программа использует данные о просмотренных страницах сайта за определенный период времени.

Настройки расширения позволяют задать следующие параметры:

- максимальный размер списка;
- максимальную длину отображаемого фрагмента названия статьи;
- включить или отключить вывод числа просмотров для каждой статьи;
- период времени, по истечении которого статистика обнуляется (день, месяц, год);
- способ определения уникальных посетителей (по IP-адресам или по пользовательским сеансам).

Период времени, для которого формируется статистика, лучше задавать таким, чтобы список самых популярных статей обновлялся достаточно часто (например, раз в месяц).

6 С помощью расширения *Most Commented* (http://dev.wp-plugins.org/browser/most-commented/) можно получить список самых комментируемых статей блога. Для каждой статьи отображается общее количество комментариев.

Настройки *Most Commented* позволяют задать максимальную длину списка статей, а также изменить формат вывода списка.

Борьба со спамом

Каждый, кто хотя бы раз администрировал мало-мальски посещаемый сайт, знает, что вместе с потоком отзывов от заинтересованных пользователей на ваши форумы, в гостевые и комментарии к записям рано или поздно приходит несравнимо больший поток сообщений от роботов, активно рекламирующих специальные лекарственные средства и услуги. *Wordpress* – не исключение из общего правила.

Можно выделить три основных типа спама, возможных в *Wordpress*.

Спам в комментариях. Самый распространенный и самый неприятный тип спама. Для борьбы с ним могут использоваться средства лексического анализа сообщений или CAPTCHA (от англ. “Completely Automated Public Turing test to tell Computers and Humans Apart” – полностью автоматизированный публичный тест Тьюринга для различения компьютеров и людей). Оба способа не лишены недостатков: расширения для автоматической фильтрации часто отсеивают довольно большую часть полезных сообщений, а для CAPTCHA любого уровня сложности при желании можно написать скрипт распознавания картинки.

Спам в трекбэках. «Трекбэк» (trackback) представляет собой способ дистанционного взаимодействия между различными сайтами. В соответствии с этим механизмом один из сайтов может уведомить другой о связи между некоторыми их частями, например, ассоциировать тематически сходные статьи. Для этого первый сайт посылает второму сообщение – «трекбэк пинг» (Trackback Ping). Получивший сообщение сайт

отправляет подтверждение в виде XML-сообщения. В *Wordpress* трекбэки для каждой статьи могут отображаться рядом с комментариями.

Возможны массовые нежелательные рассылки трекбэк-пинга. Проще говоря, спам. Для борьбы с ним разработан ряд расширений, большинство из которых используют самый простой способ борьбы с такого рода спамом: проверяют сайт-источник на наличие ссылки на ваш блог. Если на сайте-источнике такой ссылки нет, присланный трекбэк пинг помещается в очередь на модерирование.

Спам в реферерах. Большинство пользователей получают статистику блога с помощью одного из стандартных сервисов, например, Hotlog или LiveInternet.Ru. Многие оставляют эту статистику, которая содержит и список рефереров сайта, в открытом доступе. Этим могут воспользоваться спамеры, которым достаточно запустить на сайт парутройку роботов для размещения спама в комментариях, чтобы попасть в список ваших рефереров.

Следовательно, для борьбы с такого рода спамом достаточно нейтрализовать спам в комментариях.

7 Расширение *Bad Behavior* (http://www.homelandstupidity.us/software/bad-behavior/) производит фильтрацию спама всех типов. На основе анализа HTTP-запросов роботов и сравнения их с типичными HTTP-запросами пользователей сайта блокируются большинство сообщений нежелательного характера. Сравнение может осуществляться как непосредственно по строкам запроса, так и с использованием регулярных выражений.

Проанализировать статистику работы расширения и задать его настройки можно на странице Настройки – Bad Behavior.

Еще одним средством борьбы со спамом являются разнообразные генераторы CAPTCHA. Однако для изображения любого уровня сложности при желании можно написать скрипт разбора, да и для пользователей распознавать такие картинки – занятие не из приятных. Тем не менее, здесь есть и оригинальные решения, об одном из которых мы сейчас и поговорим.

8 За идею расширения *reCaptcha* (http://recaptcha.net/learnmore.html) его разработчикам можно смело ставить памятник. Вот в чем она состоит: пользователю предлагается разобрать фрагмент отсканированного из настоящей книги текста, состоящий из двух слов. Как известно, отсканированный текст очень плохо распознается компьютером. Поэтому разработчики *reCpatcha* и решили возложить эту нелегкую задачу на плечи пользователя.

Но как же проверить правильность введенного пользователям ответа? Очень просто. Одно из двух слов заранее подбирается таким образом, чтобы оно хорошо распознавалось компьютером, второе – нет. Если пользователь правильно распознает первое слово, его ответ считается правильным, и результат распознавания второго слова передается на сервер.

После этого второе слово повторно передается для распознавания нескольким пользователям, и по общему результату всех проверок принимается решение о правильном для него варианте. Таким образом пользователи, сами того не зная, вносят свой вклад в распознавание книг электронного архива Internet Archive (http://www.archive.org/index.php).

На десерт

При всех достоинствах движка *Wordpress* и автономных блогов в целом более 72% пользователей предпочитает им стандартные блог-хостинги. Причин тому множество, начиная с отсутствия специальных знаний у многих пользователей и заканчивая необходимостью платить за хостинг.

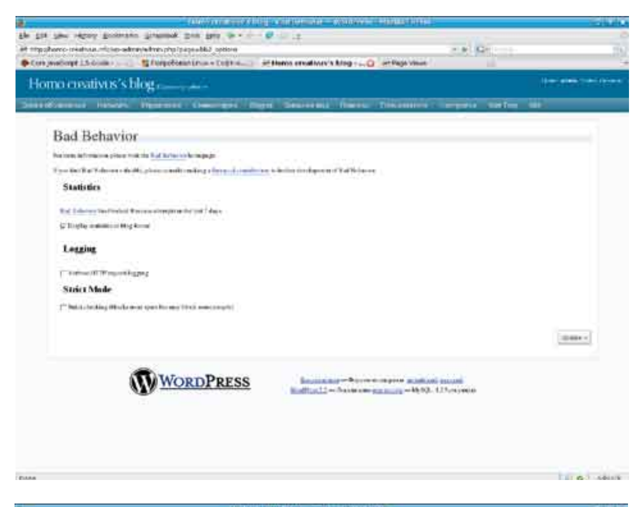
Чтобы не оказаться в изоляции от основной части блогосферы, автономный блоггер может создать собственную учетную запись в ЖЖ и как-то синхронизировать его содержимое с основным блогом.

Есть несколько вариантов такой синхронизации. Самый простой способ – скопировать запись в ЖЖ вручную после размещения ее в основном блоге. Однако каждый раз, когда в существующие записи блога вносятся какие-то изменения, эту нехитрую операцию придется повторять.

Комментарии к записям в основном блоге и в ЖЖ также не будут связаны друг с другом. Как результат, пользователи ЖЖ не увидят комментариев, оставленных в основном блоге, а читатели основного блога – комментариев в ЖЖ. Таким образом, описанная схема не только не позволяет как следует синхронизировать блоги, но и порождает огромное количество бесполезных действий.

9 К счастью, у этой проблемы есть весьма элегантное решение. Расширение *LJ Crossposter* позволяет осуществить автоматическую перепубликацию («кросспостинг») записей основного блога в блог LiveJournal. Кроме этого, поддерживаются DeadJournal (http://www.deadjournal.com/) и другие платформы на основе LJ.

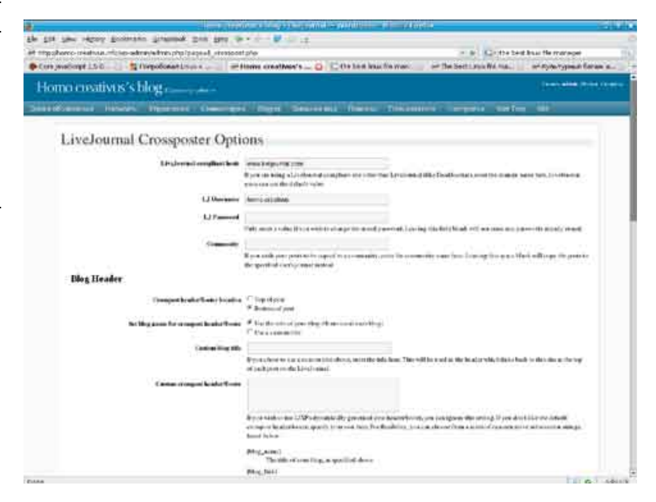
В момент размещения записи в основном блоге расширение авто-



› **Рис. 7. Расширение *Bad Behavior*.**



› **Рис. 8. Расширение *reCaptcha*.**



› **Рис. 9. Расширение *LiveJournal Crossposter*.**

матически публикует ее в ЖЖ, а в случае изменения или удаления записи соответствующие изменения также вносятся в ЖЖ.

После установки расширения на странице размещения записи появится виджет с настройками кросспостинга для каждой записи. Эти настройки позволяют включить или отключить кросспостинг записи в ЖЖ, включить или отключить комментарии к записи и задать ее видимость (публичная, только для друзей, скрытая).

Остальные настройки расширения можно найти на странице Настройки > LiveJournal. Они позволяют выбрать используемую платформу (по умолчанию LiveJournal), задать имя пользователя и пароль, а также указать сообщества ЖЖ, в которые запись также может быть автоматически переопубликована вместе с публикацией в журнале.

Одной из ключевых возможностей *LJ Crossposter* является выбор способа синхронизации комментариев в ЖЖ и основном блоге. Возможны два варианта. В первом случае комментарии не синхронизируются совсем, и в основном блоге и ЖЖ хранятся разные комментарии к одним и тем же записям.

Во втором случае все комментарии хранятся в основном блоге, а ссылки просмотра комментариев и добавления нового комментария в каждой записи ЖЖ перенаправляют читателя в основной блог. Именно по такой схеме и работают многие автономные блоги, например, дневник Александра Плющева (http://plushev.com).

Расширение позволяет переопубликовать в ЖЖ сразу все записи основного блога или записи из конкретных рубрик.

Как уже говорилось выше, если комментарии в ЖЖ и основном блоге синхронизированы, то к каждой записи в ЖЖ добавляется ссылка на эту запись в основном блоге. Внешний вид ссылки и ее расположение можно изменить, отредактировав исходный файл расширения, но одна проблема все же остается: в ЖЖ не отображается количество комментариев к записи в основном блоге.

Один из вариантов ее решения реализован разработчиками расширения *LiveJournal Comments* (http://a-bishop.spb.ru/archives/464). Оно формирует картинку с количеством комментариев к каждой записи с помощью HTTP-запроса к основному блогу. Для вставки рисунка в текст записи ЖЖ необходимо добавить вызов функции lj_comments() в код формирования заголовка записи в расширении *LiveJournal Crossposter*.

Для защиты содержимого блога от перепубликации на других сайтах могут использоваться различные механизмы. Одним из наиболее простых и эффективных способов является использование цифровой подписи. Под цифровой подписью здесь понимается некий набор символов (букв и цифр) или фраза естественного языка, уникальные для данного блога. После задания этого набора строка цифровой подписи вставляется в виде комментария в текст каждой записи и в ленту RSS. Дальнейший поиск по строке подписи в одной из поисковых систем позволит определить набор сайтов, на которых размещены статьи вашего блога.

10 Описанный механизм реализует расширение *Digital Fingerprint* (http://www.maxpower.ca/Wordpress-plugin-digital-fingerprint-detecting-content-theft/2006/09/25/). После его установки настройки можно задать на странице Управление > Digital Fingerprint. На этой странице задаются собственно цифровая подпись и ее положение в каждой записи по умолчанию (в начале записи, в конце записи, после первого абзаца). Подпись может включаться в каждую запись автоматически или вручную с помощью соответствующей кнопки на панели инструментов редактора.

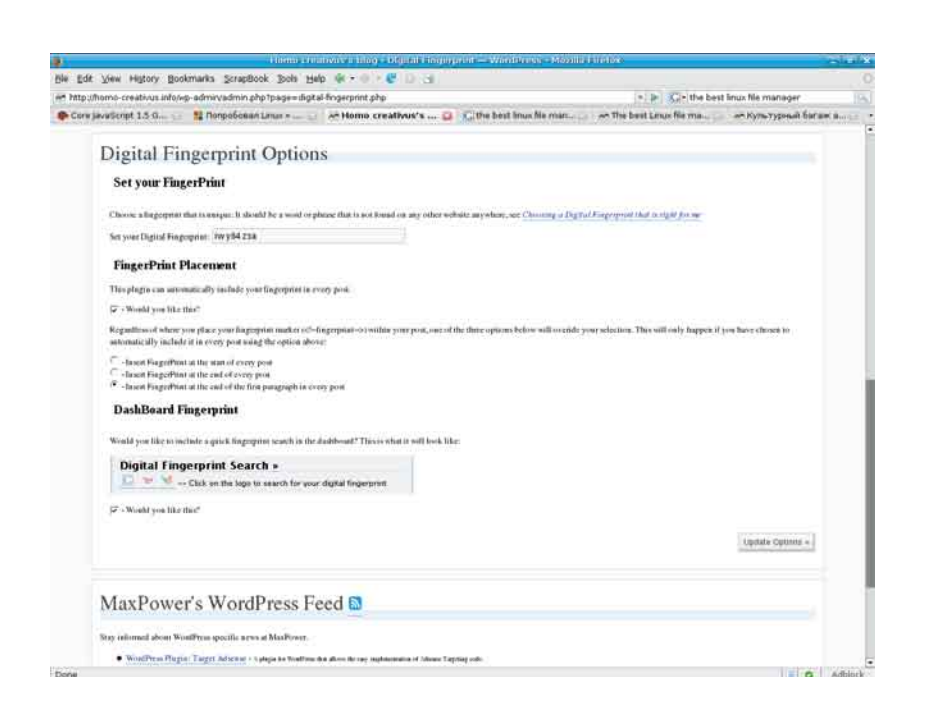
При выборе подписи необходимо руководствоваться следующими принципами:

1. Чем короче подпись, тем меньше шансов, что она будет обнаружена как людьми, так и при автоматической перепубликации записи.
2. Не следует использовать HTML, так как теги могут быть вырезаны при автоматической перепубликации записи.
3. Используйте необычные слова и фразы, которые нигде больше не встретите.
4. Перед использованием подписи проверьте ее уникальность с помощью известных поисковых систем.

Помните, что согласно теореме о бесконечных обезьянах (http://ru.wikipedia.org/wiki/Теорема_о_бесконечных_обезьянах), какую бы подпись вы ни выбрали, нет никаких гарантий того, что другой человек не выберет точно такую же. Поэтому стопроцентной гарантии это расширение не дает.

И напоследок

Мы представили вашему вниманию десятку лучших расширений для *Wordpress*, но многие интересные расширения остались за рамками нашего обзора: например, *Latest Comments*, который формирует список последних комментариев (http://maxsite.org/go/latest-comments.zip) или *WP Admin Drop Down Menus* (http://www.stuff.yellowswordfish.com/admin-drop-down-menus/), который выводит в панели администрирования выпадающее меню взамен традиционного. Думаю, вы сможете найти для себя еще много интересного. Удачи! LXF



› Рис. 10. Расширение *Digital Fingerprint*.

Мнение: Джереми Эллисон

Безумие по Эйнштейну...



Джереми Эллисон

Джереми – ведущий разработчик команды Samba, сейчас работает в Google, в Сан-Хосе, Калифорния.

В компьютерном мире стандартам ISO жилось по-всякому. Году в 1977 ISO разработала стандарт семиуровневого сетевого протокола, а он выпал в осадок, столкнувшись со стандартным протоколом TCP/IP, который стал Интернетом. Однако ISO все же стандартизировала команды UNIX и программный интерфейс POSIX, внеся элемент здравого смысла в раздробленный мир проприетарных реализаций UNIX. Похоже, компьютерные стандарты ISO работают лучше, если они берутся за нечто уже существующее.

Однако стандарты не правят компьютерным миром. На сегодняшний день 92 % настольных компьютеров и 70 % серверов работают на проприетарной и нестандартизированной ОС Microsoft Windows. Хотя POSIX является стандартом ISO, рынок склоняется в сторону Windows, являющейся стандартом де-факто. Играет роль эффект сети. Определение Microsoft Windows полностью подконтрольно его создателю, и «Windows» – это то, что решит Microsoft, ни больше, ни меньше. Исторически, отношение Microsoft к стандартам было двойственным. Приведу пример из близкой мне области протоколов обслуживания файлов: когда в этой области доминировала Novell Netware, Microsoft горой стояла за стандарты, публикуя спецификации собственных протоколов (именовавшихся тогда Server Message Block, или SMB), и поддерживала реализации на других платформах, не только на Windows. Они даже предложили протокол SMB (ныне переименованный в Common Internet File System, CIFS) в качестве стандарта Интернет (IETF).

Но как только Netware уступила позиции Windows NT, отношение изменилось: поток информации иссяк. Последовали проприетарные модификации основных протоколов, подобные аутентификации Kerberos, и все эти изменения рассматривались как коммерческие секреты, по возможности защищаемые патентами, и выходили в свет – если вообще выходили – под лицензионными соглашениями, обязывавшими пользователей к неразглашению. И именно этот зафиксированный в анналах истории факт делает недавнюю глобальную деятельность по поводу формата документов «Office Open XML» (OOXML) столь интересной. В этом случае конкуренту удалось застолбить участок первым. Открытый проект под названием OpenOffice.org (за которым пристально наблюдает Sun, ярый конкурент Microsoft) создал стандарт ISO для хранения офисных документов под названием «Open Document Format» (ODF) [открытый формат документа, – *прим. пер.*] , который был принят еще несколькими офисными программами, хотя, конечно, занимаемая ими доля рынка и рядом не стоит с таковой *Microsoft Office*. Эта ситуация опасна для Microsoft тем, что, коль скоро возник официальный стандарт ISO для офисных документов, потенциально возможно лоббирование правительств ради принятия его в качестве официального стандарта делопроизводства. А ODF – формат, в настоящее время не поддерживаемый *Microsoft Office* и не контролируемый компанией Microsoft.

В ответ Microsoft создал формат офисных документов OOXML, ставший стандартом в пакете *Office 2007* и «подарили» Европейской ассоциации производителей компьютеров [European Computer Manufacturers Association, ECMA], чтобы последняя провела его через процесс стандартизации ISO. ECMA была настолько уверена в принятии OOXML, что заявила его на ускоренное прохождение данной процедуры [«fast-track»], позволяющее стать стандартом без каких-либо изменений.

По долгу службы мне пришлось читать документацию на стандарт OOXML. При объеме в 6 000 страниц (сравните с 867 страницами стандарта ODF) это было отнюдь не тривиальной задачей. Мне нет нужды приводить здесь технические недостатки OOXML: хватает и сайтов, и блогов, детально их описывающих (например, **www.noooxml.org**). Достаточно сказать, что при таком количестве проблем, если следовать букве процедуры ISO, пригодилась бы не ускоренная процедура стандартизации, а процесс, специально разработанный для исправления технических проблем.

Чем дальше, тем любопытнее…

Однако политический охват стандартов ISO куда обширнее, чем у стандартов интернет (IETF) или Консорциума Всемирной Паутины [World Wide Web Consortium, W3C]. Голосовать может каждая страна, хотя не все они решили принять участие. Зато выявилась масса странностей и даже иррационализма в позициях некоторых стран. Лично я в восторге от проголосовавших «за» стандартизацию OOXML Кубы – хотя американское правительство запретило Microsoft продавать Острову Свободы какое бы то ни было ПО, способное работать с этим форматом – и Азербайджана: OOXML в его нынешнем виде не может писать Web-адреса на азербайджанском, их государственном языке.

Как вы знаете, «быстрое» голосование провалилось, хотя разница составила всего 1 % (это очень мало). Так что следующим шагом, согласно сложным правилам ISO, станет заседание, на котором все комиссии, участвующие в процессе, соберутся и рассмотрят технические замечания, чтобы дать дальнейший ход процессу.

ISO International Organization for Standardization

На этом заседании ECMA (и, очевидно, Microsoft, в качестве члена ECMA) придется пообещать внести изменения в формат, реализованный в *Microsoft Office 2007* и уже распространяемый в таких объемах, поставка которых по зубам только лидеру рынка. Это напоминает мне обмен email-сообщениями в тот период, когда Microsoft предпринимала попытки стандартизировать протокол совместного доступа к файлам, реализованный в Samba (CIFS). Тогда после публикации заявления, демонстрирующего атаку посредника на текущий протокол, Microsoft опубликовала спецификации для криптографической защиты обмена сообщениями, что позволило залатать дыру в системе безопасности и стало неплохой первой реакцией на атаку. Увы, после экспертного анализа обнаружилось, что новый протокол подписи имеет теоретически слабые места, и для их ликвидации и улучшения системы безопасности требуется несколько тривиальных поправок. Microsoft же ответила, что, хотя поправки и справедливы, код, к сожалению, уже написан и будет поставляться со следующим сервисным пакетом. Конец обсуждения. Код даже не был еще включен в распространяемый продукт, однако спецификацию изначально определили как нечто неизменяемое, поскольку не желали править готовую программу.

Надеюсь, что заседание по техническим замечаниям к OOXML пройдет лучше, чем оно прошло для CIFS. И еще я надеюсь, что на сей раз все может измениться. В конце концов, я не занимался бы разработкой свободно распространяемых программ/программ с открытым исходным кодом, если бы не считал, что все может измениться, как, например, 92 процента рынка настольных систем или 70 процентов рынка серверов. Или, может быть, это просто делает разработчиков свободного ПО настолько невменяемыми, насколько их пытаются выставить таковыми поставщики проприетарных решений?

В конце концов, как утверждают, Эйнштейн однажды сказал: «Безумие – это когда постоянно совершаешь одно и то же действие и каждый раз ожидаешь разных результатов». LXF

Думая об ISO

До недавнего времени я не особо вникал в стандарты Международной Организации по Стандартизации [International Standards Organization] (ISO); они касаются самых разных вещей, от, например, труб (стандарт ISO номер 7) до размеров обуви (стандарт ISO 407) и скорости кинопленки (ISO 5800). И хотя они незримы для простого человека, современная жизнь намного упрощается благодаря их международному принятию; только представьте себе сложности при покупке китайских туфель, если бы изготовители не придерживались стандарта размеров.

LINUX FORMAT

Учебники



Наши эксперты помогут вам с любым приложением Linux



ЕВГЕНИЙ БАЛДИН
Начинал с Агатов. Когда-то даже знал, что такое Робик.

Мастера или ремесленники?

Фраза «Стране нужны кодеры, программистов и так хватает» – все равно, что утверждение: «Стране нужны Донцовы, Пушкиных и так хватает».

Из разговоров на LOR

Естественно, и кодеры, и «Донцовы» нужны. Особенно «Донцовы», потому что быть «Донцовой» весьма непросто – много сложнее, чем кодером. Тем более, Пушкин – это, фактически, первый на Руси коммерческий писатель (какое счастье, что ему не хватало денег на игру в карты, и он вынужден был временами работать).

Основная проблема в следующем: для того, чтобы могла работать «Донцова», до нее должен быть Пушкин. Аналогично, именно программист дает возможность работать кодерам, а вовсе не наоборот. Более того, он создает новые рабочие места там, где их раньше не было.

Допустим, что с пользой от существования, скажем так, «творцов от программирования» никто не спорит. Но как им стать? Первый шаг очевиден: учеба. Причем обучаться следует фундаментальным вещам, то есть, как минимум, математике. По-хорошему, зная математику, следует разбираться и в какой-либо предметной области, которая вас интересует. «Добрать» такого рода знания потом нереально. Фундамент требует времени для закладки, причем делать все это нужно основательно.

А в качестве практики проще всего выбрать один из свободных проектов. В этом случае плохой код спрятать не удастся – он виден всем, поэтому придется делать все «сразу правильно», и это может стать хорошей привычкой. Ее можно считать за второй шаг.

E.M.Baldin@inp.nsk.su

В этом выпуске...



Совет месяца: Смена редактора



Если вы – пользователь дистрибутива на основе Debian (например, Ubuntu), то наверняка обращали внимание на загадочный каталог **/etc/alternatives**. Заглянув в него, вы обнаружите огромное количество стандартных системных команд. Взглянув попристальнее, вы поймете, что все это – символьные ссылки на реальные исполняемые файлы в вашей системе. Скопище ссылок обязано своим существованием разработчикам Debian, которые решили не делать за вас предпочтений в выборе инструментов. Для иллюстрации своей позиции они обычно используют *cron*, демон-планировщик. Чтобы назначить некоторое событие на определенные дату и время, *Cron* запускает текстовый редактор, в котором вводится описание задачи. Вопрос, утверждают разработчики Debian, в том, какой именно редактор? Linux не дает на него однозначного ответа, и «правильный» текстовый редактор послужил предметом бурных споров. Предпочитает ли пользователь *Emacs*, *Vi* или *Nano*, жесткий выбор одного из них всегда приводит к проблемам.

Решением стал **/etc/alternatives**. Набрав *cron* в Ubuntu, вы увидите дружелюбный к новичкам *Nano*. На самом деле, cron просто вызывает команду editor, расположенную в **/usr/bin** и являющуюся ссылкой на **/etc/alternatives/editor**.

Ну, а последняя, как вы уже поняли, указывает на реальное приложение – в нашем случае, **/usr/bin/nano**. Чтобы изменить это поведение в соответствии со своими предпочтениями, достаточно заменить ссылку editor в **/etc/alternatives**. Существует даже команда, готовая выполнить эту работу за вас. Введите update-alternatives --set editor /usr/bin/vim, тогда редактором станет *Vi*. Для получения перечня возможных вариантов, используйте ключ -display editor. Точно так же можно поступить с любой командой, определенной в **/etc/alternatives**.

Настольные ПК:

Старый ПК пылится в углу? Снабдите его не жадным до памяти дистрибутивом Linux и используйте как сетевой суперкомпьютер. **Энди Ченнел** снова в 1999...





Наш эксперт

Энди Ченнел
Энди делает свои первые шаги в Linux уже шесть лет, а технологиями интересуется еще со времен Dragon 32

Gnome и KDE практически полностью захватили рынок рабочих столов Linux, но вовсе не обязательно ограничивать выбор лишь этими двумя вариантами. Более того, если ваш компьютер старше пяти лет, универсальность как KDE, так и Gnome способна удручающе повлиять на эффективность его работы. А если ваши потребности не заходят далее прогулок по сети, операций с почтой и несложной обработки текстов – или вы решили справить второй компьютер сугубо для этих целей – небольшие дистрибутивы Linux могут уберечь вас от хлопот и чрезмерных расходов на вылазку в ваш локальный компьютерный мирок.

На данном уроке мы представим трио дистрибутивов Linux, с которыми снова запоют ваши старые устройства и которые, в связке с широкополосным подключением и бесплатной сетевой магией, подарят вам полный функционал рабочего стола, даже если вы не наскребете для него исправный жесткий диск. Этот второй аспект, кстати, полезен и за рамками урока, так как означает, что вы можете спокойно вести цифровую жизнь только на CD- или USB-диске и получать доступ к знакомому окружению и всем вашим документам практически с любого компьютера, подключенного к Интернету. Опять же, дистрибутивы, чей девиз – неприхотливость, бережнее расходуют энергию, а стало быть, снижают ваше воздействие на окружающую среду. Познакомившись с двумя очень маленькими дистрибутивами, мы сфокусируемся на Fluxbuntu, который предлагает великолепную комбинацию скромного размера и удобства знакомых инструментов рабочего стола.

Однако учтите: мы не будем интересоваться великолепием видеоэффектов, мощью мультимедиа или отпадной производительностью игр; они тоже входят в каноны Первых Шагов, но сейчас для нас главное – малый размер, переносимость и доступность. И это прекрасные цели.

Рассматриваемые ниже дистрибутивы будут как дома на старой машине Pentium II с тактовой частотой где-то 400–600 МГц и ОЗУ 128 МБ. Мы также полагаем, что у вас имеются работающие графическая и звуковая подсистемы, быстрое интернет-соединение и, при желании установить дистрибутив, жесткий диск с парой свободных гигабайт. Можно, конечно, приискать дистрибутивы Linux и для совсем убогих машин, но мы не собираемся переходить к работе с командной строкой, так что это наш минимум.

Что до самих окружений рабочего стола, то у разработчиков Linux имеются два основных инструментария для создания графических приложений: это *Qt*, основа KDE, и *GTK+*, используемый в Gnome. В смысле выбора это здорово, но если вы пользователь Gnome, желающий создавать документы в *KOffice*, кроме приложения вам придется загрузить библиотеки KDE, а тогда – прощай, экономия от малых дистрибутивов. Как следствие, большинство подобных проектов базируется на одном инструментарии – обычно *GTK+* – и вы не найдете здесь *OpenOffice.org*, *Konqueror* или *Amarok*. Первый из выбранных нами дистрибутивов вообще избегает обоих инструментариев Linux.

Damn Small Linux

Начнем мы с Damn Small Linux (DSL), короля малых дистрибутивов. DSL разрабатывался с прицелом на не больно-то распространенные CD размером с визитку, и потому ограничил сам себя размером 50 МБ. На всю систему! DSL – подлинное воплощение «малолитражного» Linux; разработчики даже остановились на ядре 2.4, чтобы снизить его объем. В результате вы запросто запустите DSL на компьютере 486DX с жалкими 16 МБ ОЗУ, а уж если у вас 128 МБ, то система сможет работать целиком в оперативной памяти, достигая превосходной производительности даже на антикварном оборудовании. Мы использовали

Дистрибутивы-крошки

Выбирайте легковесный дистрибутив с той же осторожностью, что и основной. К счастью, у пользователей Linux появился новый помощник в поисках идеального дистрибутива: виртуализация. Чтобы просто опробовать новый дистрибутив, больше не нужно вычищать вашу рабочую машину и устанавливать его – это если у кого есть рабочая машина – или дробить разделы диска основной машины. Теперь, благодаря различным свободным и коммерческим Linux-технологиям виртуализации, проще создать для тестирования виртуальную машину (VM), не затрагивая другие части вашей системы, а потом ее удалить. Вы можете даже найти в сети готовую VM специально для вас, загрузить ее и наслаждаться самым простым в использовании из бесплатных приложений виртуализации: *VMware Player*.
www.vmware.com/products/player/

Месяц назад Подправляли Gnome: и визуально, и функционально.



легкие дистрибутивы

последний (на момент написания) релиз-кандидат DSL 4.0, загрузив его с www.damnsmalllinux.org. Размер загружаемого ISO-файла всего 47 МБ – не убийственно даже при закачке по телефонной линии.

Привыкшим к SUSE, Ubuntu или Red Hat покажутся волшебством считанные секунды на загрузку DSL, а стандартный рабочий стол имеет все необходимое для базовых операций. Внизу экрана находится панель задач, где, слева направо, располагаются: кнопка пуска (помеченная как DSL); кнопка сворачивания, которая все прячет, чтобы вы могли видеть рабочий стол; квартет переключателей рабочих столов, запущенные приложения и часы. По умолчанию дистрибутив стартует, открывая web-браузер Dillo, и, при наличии сетевого подключения, вы можете набрать в его строке адрес и начать путешествие по сети.

Если, вставив CD и включив машину, вы увидели свою старую ОС или вообще ничего, значит, ваш ПК не настроен на загрузку с CD (очень древние машины могут даже не иметь этой опции); тогда вам придется об этом позаботиться. Перезапустите компьютер и при старте нажмите клавишу для входа в Setup или BIOS – обычно это F2 [или Del, – *прим. ред.*]; стартовый экран содержит сообщение вроде “*Press F2 to enter set up*” [*«Нажмите F2 для входа в настройки»*]. Оказавшись в меню, найдите параметр с названием типа Boot order [Порядок загрузки] и измените его. Традиционно устанавливалась такая очередность: Floppy Disk, Hard Disk, CD; мы пойдем другим путем, чтобы машина грузилась с CD или, не найдя загрузочного диска в приводе, возвращалась к установленной на жестком диске. А вот валяющиеся повсюду старые ноутбуки Toshiba загружаются с CD, если после включения удерживать клавишу C. Загрузка с USB-диска, допускаемая всеми нашими дистрибутивами – явление новое, и вряд ли будет поддерживаться на старых машинах. Можно, конечно, заставить старую машину загрузить Live-дистрибутив с USB-диска при помощи загрузочной дискеты, но это вне компетенции учебника **Первые шаги**.

Очень скоро вы заметите, что хотя с простыми сайтами все в порядке, Dillo, стандартный web-браузер DSL, не обрабатывает многие элементы оформления вроде CSS, поэтому он не подойдет для работы с текущим набором сетевых приложений. К счастью, малость устаревшая версия *Firefox* спрятана в DSL > Apps > Net > Browsers > Firefox. Хоть это версия 1, она позволит вам заходить на сайты вроде iGoogle, Zoho и Netvibes, не открывающиеся в *Dillo*.

Справа на экране помещены системные мониторы, показывающие, насколько DSL экономичен, сколько занято места в ОЗУ/на диске, IP-адрес компьютера и время работы системы. Слева находятся четыре иконки, предоставляющие доступ к (сверху вниз) Home [домашний каталог], Root [Корневой раздел], Applications [Приложения] – для фанатов Windows 3.1 – и MyDSL. MyDSL позволяет добавлять в систему приложения и службы при помощи простого интерфейса, с разумно подобранными настройками. Правда, чтобы получить от этого пользу, необходимо установить дистрибутив на диск.

DSL спроектирован для запуска как LiveCD, и именно так он работает лучше всего, но если у вас есть жесткий диск с парой сотен свободных мегабайт, то имеется также возможность установки, хотя и не рекомендованная для «рабочих» машин: другими словами, используйте это на свой страх и риск и не удивляйтесь, если обновление на DSL не будет работать корректно с установленной системой. Установка пока поддерживается только на английском языке. Для начала надо создать



› DSL мало что предоставляет сверх минимума, но если ваши нужды или техника скромны, оно и хорошо.

раздел по крайней мере в 200 МБ для размещения ОС. Выполните DSL > XShells > Root Access и выберите одну из оболочек. При появлении терминала, наберите cfdisk /dev/hda (где hda – имя вашего жесткого диска), и используйте опции создания нового основного [primary] раздела, являющегося загрузочным (этот выбор предоставляется во время процесса). Запишите имя диска (в нашем случае, hda1), поскольку именно сюда мы собираемся установить DSL.

Для установки DSL, выберите DSL > Apps > Tools > Install to Hard Drive и, по запросу, выберите раздел для установки – **/dev/hda1**, или что-то еще – а затем следуйте подсказкам. Процесс протекает очень быстро: вас попросят подтвердить, что вы хотите перезаписать созданный ранее раздел и настроить загрузчик (*Grub* или *Lilo*; мы выбрали первый), и затем компьютер выключится и перезагрузится. Последний этап – выбор нового пароля root; учтите; DSL не примет слабый пароль, так что придумайте что-то длиннее шести символов, включающее цифры и буквы в верхнем и нижнем регистре, и процесс будет завершен. Теперь все ваши данные будут храниться в **/home**, как и в обычном дистрибутиве. После установки мы заметили, что DSL использует лишь 30 % доступной оперативной памяти и лишь половину нашего массивного диска в 256 МБ; похоже, резервы есть. Пора обновляться...

DSL-Not

Если у вас ресурсов чуть больше, можно получить более элегантное окружение и не столь сталинский минимализм, благодаря ответвлению DSL, Damn Small Linux – Not. DSL-N вынул ядро дистрибутива и усилил его новым, серии 2.6, и лучшим определением устройств. Он также использует инструментарий GTK2, сделав доступным более широкий диапазон приложений, в том числе такой оплот FOSS, как *AbiWord*, для »

Скорая помощь



Настройки даты и времени можно синхронизировать с серверами времени в Интернете, чтобы они всегда были точны. Gnome предлагает для подключения набор серверов времени, включая один в Великобритании и несколько – в России.

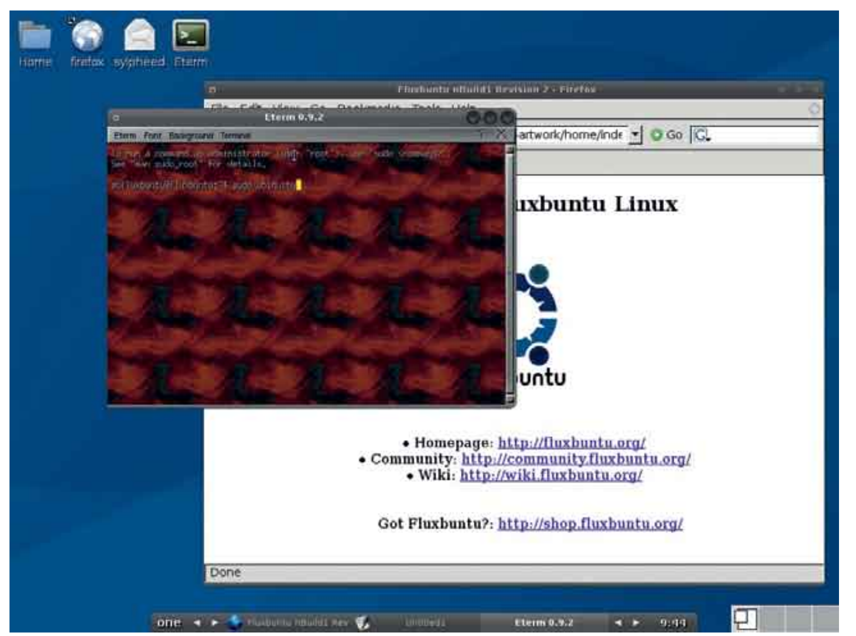
» обработки текстов на мировом уровне, и GIMP для редактирования изображений. Оборотная сторона в том, что объем дистрибутива 95 МБ (доступен для загрузки на www.damnsmalllinux.org/dsl-n/), и он требует чуть больше, чем 16 МБ ОЗУ, но это хороший компромисс между экономностью и удобством. Здесь даже предлагается выбор рабочего стола: JWM или Fluxbox. Первый быстрее, зато второй более элегантен. Установка DSL-N в точности аналогична прародителю и показала ту же скорость на нашей старой системе, а улучшения стоят хлопот. Однако он все еще не до упора загрузил наше оборудование, и мы можем поискать нечто более стоящее. Время для очередного обновления...

Fluxbuntu

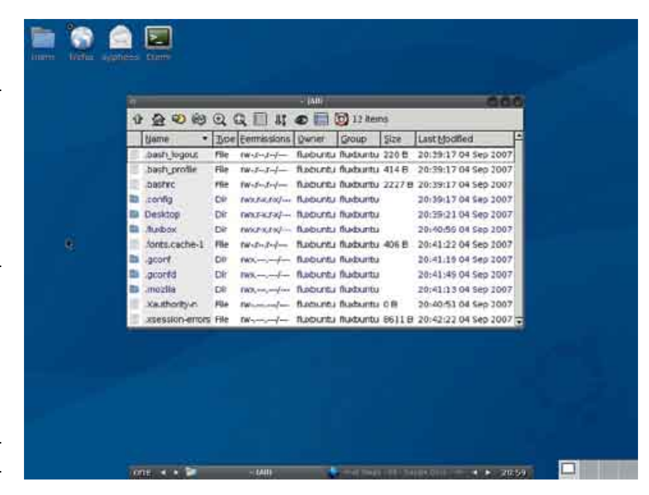
Следующий в нашем списке и еще один шаг по шкале эволюции – это Fluxbuntu, который, как следует из названия, является версией Ubuntu, созданной вокруг оконного менеджера Fluxbox. LiveCD Fluxbuntu занимает более 300 МБ – в противовес 700 МБ его старшего брата – и предоставляет сходный набор приложений, охватывающий все. Вы можете загрузить последнюю версию с www.fluxbuntu.org, и хотя имеется только версия для процессоров x86, поддержка PPC и 64-битных Intel/AMD вариантов находится в разработке. Как и в двух предыдущих случаях, Fluxbuntu поставляется в виде готового LiveCD: необходимо только скачать и прожечь его образ и загрузиться с диском в приводе.

Сделав это, вы заметите знакомые (многим) загрузочные окна Ubuntu, а кто знаком с начальной загрузкой CD Ubuntu, заметит также, что Fluxbuntu резвее; на нашей тестовой машине загрузка Fluxbuntu заняла половину времени для стандартного диска Feisty Fawn, и это хорошая новость, если вы планируете использовать его как тонкий клиент для подключения к сетевым службам типа Zoho. А еще вы заметите, что Fluxbuntu превосходно определяет устройства, благодаря своему наследию, то есть все ваши устройства будут готовы к использованию сразу же после загрузки дистрибутива.

В отличие от первых двух дистрибутивов, перед тем как что-то сделать, вам потребуется войти в Fluxbuntu; стандартное имя пользователя fluxbuntu, а пароль livecd. Сразу после входа перед вами предстанет красивый рабочий стол, снабженный четырьмя иконками слева вверху (для доступа к покамест бесполезному домашнему каталогу, *Firefox*, почтовому клиенту *Sylpheed* и всегдашнему терминалу). Доступ к другим приложениям, включенным в дистрибутив, осуществляется правым щелчком в любом месте рабочего стола и выбором требуемого приложения или места в всплывающем меню; точно так же, как использование панели в KDE, Gnome или даже Windows, только вызов немного отличается. У нижнего края экрана также имеется док-область и квартет виртуальных рабочих столов внизу справа. Вы можете быстро перемещаться между этими окнами, или нажимая клавиши влево-вправо, или, если у вас новая мышь, используя колесо прокрутки.



› Установка Fluxbuntu не сводится к нажатию кнопки – но это не слишком заумная задача.



› *ROX Filer* – хороший файл-менеджер, но в нем надо освоиться, если вы привыкли к *Nautilus* или *Konqueror*.

Для начала просто щелкните на иконке *Firefox* на рабочем столе и подождите, пока запустится браузер. После запуска иконка и панель задач приложения появятся в доке внизу экрана; их можно использовать для переключения между открытыми приложениями или окнами так же, как в KDE или Gnome. Вдобавок правый щелчок на этих панелях задач приводит к вызову контекстного меню для конкретного окна. Здесь имеется несколько очень полезных опций, способных реально влиять на вашу работу. Давайте пробежимся по ним сверху вниз.

» Shade сворачивает все открытые окна в строку заголовка. Прекрасно для быстрого сворачивания множества окон.

» Stick закрепляет текущее окно на экране: оно останется видимым, даже если вы будете перемещаться между виртуальными рабочими столами. Незакрепленные окна связаны со своим виртуальным рабочим столом и потому при таком перемещении исчезают.

» Maximise заставляет выбранное окно заполнить весь экран. Тот же эффект достигается при щелчке на средней кнопке в правом верхнем углу любого окна.

» Iconify минимизирует окно в панель задач. Вариант выполнения – щелкнуть на панели задач любого открытого окна или на крайней левой кнопке в правой части строки-заголовка.

» Raise перенесет выбранное окно на верхний уровень структуры, то есть можно найти, что вы ищете, даже если у вас целый стог открытых окон.

» Lower выполняет обратное предыдущему.

» Send To… позволяет отослать текущее окно на любой виртуальный рабочий стол. Нужный рабочий стол позволит выбрать всплывающее меню.

» Layer… Более продвинутая версия опций Raise и Lower, описанных выше.

» Remember… заставляет Fluxbuntu запомнить позицию, рабочий стол и размеры текущего окна; например, можно устроить так, чтобы ваш почтовый клиент всегда открывался на Рабочем столе 3, заполнял весь экран и возвышался над всеми остальными окнами.

Как и DSL, Fluxbuntu разрабатывалась как Live-система. В дистрибутив включен установщик (для его запуска, просто откройте терминал и наберите *sudo ubiquity*), но в нашем случае он часто падал. Лучше использовать механизмы LiveCD и сохранять документы на внешний диск или, еще лучше, хранить все в сети.

Стоит также рассмотреть имеющиеся приложения: например, *AbiWord* предоставляет почти весь функционал *OpenOffice.org* и *Word* – ну, по крайней мере, полезный – без ненужного хлама.

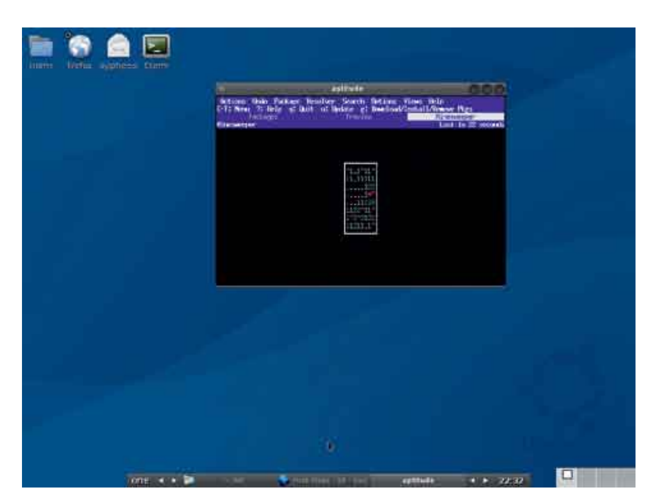
Тем, кто управлял файлами при помощи *Konqueror* или *Nautilus*, к менеджеру файлов Fluxbuntu придется попривыкнуть, но вы уж потрудитесь: все в общем логично. Для открытия файл-менеджера *ROX*

Filer, или щелкните на иконке Home, или выполните правый щелчок на рабочем столе и выберите Apps > System > ROX Filer. Интерфейс пользователя *ROX Filer* – в отличие от других файловых менеджеров – состоит из одной панели с таблицей иконок, отображающих файлы или папки. Однократный щелчок на папке откроет ее (после чего открытая папка займет все окно), а выбор файла приведет к его выполнению или запуску соответствующего приложения. Углубившись в файловую систему, вы можете вернуться, используя стрелку вверх у левого края, или вернуться в домашний каталог, нажав иконку Home, вторую слева. Чтобы скопировать файл из одного места в другое, следует открыть два окна и перенести файл из одного в другое, или использовать контекстное меню правой кнопки для копирования [Copy] файла и последующей вставки в новом месте.

Другие кнопки в верхней части окна *ROX Filer* позволяют обновить содержимое, вывести больше деталей, переключившись в режим списка, показать или спрятать системные файлы (как и в любой Unix-системе, имена спрятанных файлов начинаются с точки, т.е. **.file**) или быстро переместиться к ранее определенным закладкам.

И наконец, можно добавить новые приложения в вашу систему Fluxbuntu (после установки, конечно), используя *Aptitude*: правый щелчок > Applications > System > Aptitude. Как и другие элементы этой ОС, *Aptitude* лишен графических излишеств, присущих другим системам, но с работой справляется.

Хотя *Aptitude* и выглядит как сложное приложение командной строки, он реагирует на щелчки мыши. Чтобы начать работу, щелкните на меню View и выберите Packages. Основная панель окна изменится, отобразив набор опций. Для просмотра доступных приложений, щелкните в разделе Not Installed Packages [Неинсталлированные Пакеты] и нажмите клавишу Enter. Этот список должен раскрыться и показать различные разделы приложений, включая Graphics [Графика], Editors [Редакторы], Mail [Почта], и т.д. Воспользовавшись клавишами управления курсором Вверх и Вниз для выбора каждой из них, вы увидите описание типа приложений, находящихся в каждом разделе ниже списка. Выберите любой из этих разделов клавишей Enter, и перед вами предстанет список всех доступных программ. Например, в разделе Graphics находится *GIMP*, пакет с открытым кодом для редактирования изображений. Выберите его, нажав клавишу Enter, затем откройте меню Package и выберите Install. На самом деле это не приведет к началу процесса инсталляции, а лишь пометит приложение для установки. Теперь вы можете пробежаться по спискам и выбрать все, что вам нужно для охвата ваших компьютерных нужд. Как только будете готовы, нажмите в панели меню пункт Action [Действие] и выберите Install/remove packages [Установить/удалить пакеты]. Теперь Debian-сердце дистрибутива разберется с зависимостями для каждого выбранного вами приложения, все их загрузит (лучше не пытаться делать это по телефонной линии) и установит в систему, готовые к использованию.



› Может, *Aptitude* и не первый красавец среди менеджеров пакетов, но в нем хорошая встроенная версия *Minesweeper*.

Создаем сетевой суперкомпьютер

Как легко представить, у команды LXF целый склад старых машин – включая архаичный Athlon 2400+, на котором готовился данный урок – но что вам сгодится для создания своего собственного сетевого компьютера? Забавы ради мы откопали в чулане Pentium 60, подключили к нему все необходимые кабели и загрузили.

Эта машина в свое время считалась мощной мультимедийной станцией; в ней 64 МБ ОЗУ, 800-МБ жесткий диск (сломан), CD-ROM, интегрированная видеокарта Intel, Soundblaster и убогая стандартная сетевая карта. Как и ожидалось, DSL и DSL-N прекрасно заработали, загружаясь быстрее «родной» для этого ПК Windows 98SE; они также легко подключили все (очень) зрелые внешние устройства, и вскоре мы бродили в сети как в 1999-м. Скорость рендеринга в *Firefox* было не блестящим – создавалось впечатление, что используется подключение по медленной телефонной линии – но терпимым. Хранение данных, как и приложения, базировалось на web-сервисах, то есть отсутствие жесткого диска не стало трагедией, как могло бы. Fluxbuntu оказался более медлителен при запуске, но будучи загружено, все и тут работало как часы. Собственно, кабы не раздражающий шум вентилятора и странный запах, машина могла бы стать прекрасным дополнением к детской.



› **Даже если не работает жесткий диск, дистрибутив LiveCD сделает ПК вновь полезным.**

Тот же тест мы провели с пятилетним ноутбуком Toshiba 1.2 ГГц Celeron, доказав, что старое оборудование все еще отлично годится для работы в сети, поскольку сетевые приложения вроде *Zoho Writer* и *Box* (для хранения) работали не хуже, чем на современной двухъядерной машине, даже когда ОС загружалась с CD. А поскольку оба этих ПК уже считаются дедушками, вы, вероятно, сможете раздобыть нечто подобное за пару фунтов на eBay, а то и вообще бесплатно на Freecycle.

X-фактор – Xubuntu

Следующий шаг вверх – который мы оставляем вам – Xubuntu: тот же Ubuntu, но с рабочим столом Xfce. Этот вариант приближен и к Gnome, и к KDE по части удобства, но более жаден до ресурсов, чем три рассмотренных здесь дистрибутива. Если у вас современная машина и вам не нужны Jay-Z уровни роскоши рабочего стола, то Xubuntu – прекрасный выбор. Перейдите к DVD-разделу журнала – он начинается на стр. 104 – и просмотрите простое руководство по установке Xubuntu от Майка Сондерса. LXF



› **Fluxbuntu использует зрелое воплощение проекта *Firefox* (версия 1.5), что означает лучшую поддержку Web 2.0.**

» **Через месяц** Переезжаем в Linux из Windows, не теряя данные и настройки.

Эксперименты

Вы можете и не подозревать о существовании аутентификации, потому что она «просто работает». А вот как она работает? **Д-р Крис Браун** расскажет.



PAM (Pluggable Authentication Modules – подключаемые модули аутентификации) – один из тех темных закоулков Linux, куда большинство пользователей заглянуть не решается. Я даже готов побиться об заклад, что большинство пользователей не знают, что это такое. Однако PAM – в сердце каждой программы, так или иначе связанной с аутентификацией. Добро пожаловать на экскурсию по PAM! Заходите в нашу научную лабораторию, выполните наши эксперименты (горелка Бунзена не потребуется!), и вы увидите, что PAM предоставляет тонкий контроль над политикой безопасности.

В LXF83 была статья Грэма Моррисона, в которой он провел обзор PAM и продемонстрировал ее гибкость, показав, как с ее помощью расширить аутентификацию пользователя в Linux, используя цифровой ключ, записанный на USB-брелок. В этом месяце мы пойдем дальше и рассмотрим более существенное повышение безопасности системы с помощью PAM. Те из вас, кто настриг LXF83 на мелкие кусочки и использовал их как подстилку в клетке своего хомячка, и те, кто просто не покупал этот номер, не переживайте: в первом разделе мы проведем небольшой обзор PAM, а в детали углубимся во втором.



Наш эксперт

Д-р Крис Браун внештатный консультант по Linux со степенью доктора наук по физике элементарных частиц и обладатель сертификатов Novell CLP и Red Hat RHCE. Он только что написал книгу по SUSE Linux для O'Reilly.

Часть 1: Знакомимся с PAM

PAM – это каркас, помогающий приложениям выполнять то, что я называю «действиями, связанными с аутентификацией». PAM состоит из библиотеки (*libpam*) и набора модулей PAM, которые находятся в папке **/lib/security** и представляют собой динамически загружаемые библиотеки (.so). Каждый модуль служит для решения определенной задачи, и приложения, связанные с PAM, обычно используют стек из нескольких модулей. Общая архитектура показана на рис. 1. PAM определяет четыре типа действий, относящихся к аутентификации:

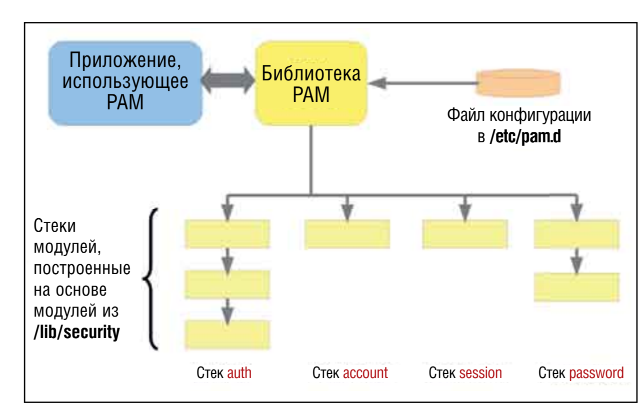
» **auth** – наиболее очевидное действие: вы удостоверяете свою личность, предоставляя некоторый набор реквизитов. Обычно это имя пользователя и пароль, но возможны и другие варианты: например, физический ключ (как в LXF83), а то и биометрические методы, вроде отпечатков пальцев или сканирования сетчатки глаза.

» **account** – действия, определяющие, можно ли вам зайти в систему (теперь, когда мы уже знаем, кто вы). К этой категории можно отнести, например, модуль PAM, ограничивающий вход пользователей в систему в зависимости от времени суток.

» **session** – действия по выделению ресурсов, которые могут потребоваться пользователю во время сессии, например, монтирование домашнего каталога пользователя, установка лимитов на использование ресурсов системы, вывод ежедневного сообщения и т.д.

» **password** – действия по обновлению реквизитов пользователя (обычно пароля).

Для краткости в дальнейшем я буду называть эти действия «задачами PAM».



› **Рис. 1. Архитектура PAM и взаимосвязь ее частей.**

Какие программы используют PAM?

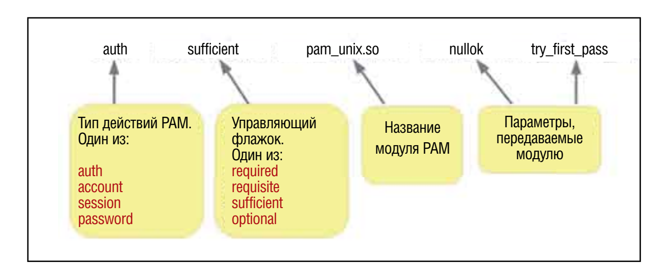
Да любые, которым нужно аутентифицировать пользователей, управлять их входом в систему, выделять им ресурсы или обновлять реквизиты: использование PAM значительно облегчает им жизнь. Очевидные примеры:

» *login* – программа входа в текстовый терминал.
» *gdm* – вход в графический рабочий стол.
» *su* – запуск оболочки от имени нового пользователя.
» *passwd* – задание нового пароля.

Преимущество PAM в том, что «действия, связанные с аутентифика-

» **Месяц назад** Диагностика ваших сетевых проблем – не обуза.

с PAM



› **Рис. 2: Синтаксис файла конфигурации PAM.**

Использует ли программа PAM?

Как узнать, использует ли PAM та или иная программа? Ну, первый способ – посмотреть на файлы конфигурации в **/etc/pam.d**. Теоретически это подскажет нам ответ. Способ похитрее – посмотреть, скомпонована ли программа с библиотекой PAM. Это можно сделать с помощью следующей команды:

```
$ ldd /bin/login | grep libpam
        libpam.so.0 => /lib/libpam.so.0
(0xb7f47000)
        libpam_misc.so.0 => /lib/libpam_misc.so.0
(0xb7f43000)
```

если вывод похож на этот, программа использует PAM. Если вывод отсутствует – нет.

цией», не привязаны к приложению, которое их использует. Благодаря этому гораздо проще менять политику безопасности или добавлять новые механизмы аутентификации: достаточно просто перенастроить PAM. Раньше для этого могло потребоваться изменение исходного кода самих приложений.

Стек модулей, с помощью которых каждое приложение с аутентификацией через PAM выполняет одно из четырех соответствующих действий, определяется в файле конфигурации приложения, расположенном в каталоге **/etc/pam.d**. Имя файла соответствует имени приложения: например, файл **/etc/pam.d/sshd** содержит настройки PAM для программы *sshd*. На рис. 2 показана одна из строк такого файла. Эта строка добавляет к стеку auth модуль pam_unix.so. При необходимости приложение может сформировать все четыре стека (по одному для каждого из типов действий PAM), однако для большинства приложений достаточно двух.

Итак, основная идея такова: когда приложению надо выполнить одно из действий PAM, оно обращается к библиотеке PAM. Затем библиотека по очереди вызывает все модули в этом стеке. Каждый модуль «делает свое дело» и возвращает библиотеке положительный или отрицательный результат. Библиотека объединяет результаты модулей в результат для стека в целом, который возвращается приложению. Способ объединения определяется управляющим флажком модуля, прописанным во втором поле строки в файле настроек. Возможные значения флажка:

» requisite Если модуль завершается с ошибкой, PAM немедленно возвращает приложению ошибку, другие модули стека не вызываются.

» required Если модуль завершается с ошибкой, PAM возвращает ошибку приложению, но продолжает вызывать остальные модули стека. (Кто не в силах запомнить разницу между requisite и required, вступайте в спецклуб! Я уже в нем...)

» sufficient Если модуль завершается успешно, PAM возвращает приложению результат «выполнен», и остальные модули стека не вызываются. (Конечно, предполагается, что никакой модуль с флажком required выше в стеке не отказал.)

» optional Результат работы модуля (выполнен/ошибка) игнорируется. Обычно это означает, что модуль вызывается для выполнения некой операции, не имеющей отношения к принятию решения «пропустить/отказать» для всего стека. Например, модуль pam_keyinit – «необязательный» модуль для *sshd*, он создает новый «ключ сессии» для нового пользователя.

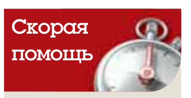
(Последние версии PAM поддерживают получение более подробного результата выполнения модуля, чем просто «выполнен/ошибка», и более сложный синтаксис для задания способа объединения отдельных результатов. Подробности можно найти на man-странице **pam.conf**, если она есть. Однако и этого простейшего отчета вполне достаточно.) В качестве управляющего флажка также может использоваться ключевое слово include. Эта ситуация немного отличается от других. Она предписывает PAM подключить некоторый предопределенный стек модулей (в терминах программирования это можно назвать вызовом подпрограммы). Это позволяет «вынести за скобку» наиболее общие действия PAM (например, общие стеки модулей), в отдельный файл. Например, в RHEL5 вы заметите, что во многих файлах конфигурации PAM подключен стек, определенный в файле **system-auth**. В Ubuntu 7.04 имеются четыре файла для аналогичной цели: **common-auth**, **common-account**, **common-session** и **common-passwd**. В Red Hat также есть своего рода «метамодуль» pam_stack, который выполняет стек модулей, определенных в отдельном файле.

Теперь, получив представление о том, как модули PAM связаны друг с другом, вы, возможно, захотите взглянуть на большую таблицу в конце этой статьи, где описаны модули, используемые чаще всего. Учтите, что некоторых модулей может не быть в вашем дистрибутиве, и, наоборот, вам могут попасться модули, которых нет в этой таблице. Поскольку Linux – открытая система и прочее, ребята понаписали всяких-разных модулей PAM. Некоторые ссылки можно найти на странице http://www.kernel.org/pub/linux/libs/pam/modules.html.

Ниже приведены шесть экспериментов с изменением настроек PAM. Ни один из них сам по себе особенно не впечатляет, но я надеюсь, что они придадут вам достаточно уверенности в себе и понимания работы PAM, чтобы начать свои собственные эксперименты. »

Документация PAM

Возможно, вам придется немного покопаться, чтобы найти документацию на PAM, потому что в разных дистрибутивах она хранится в разных местах. Например, в RHEL5 man-страницы для модулей PAM входят в основной пакет PAM; в Ubuntu документация находится в каталоге **/usr/share/doc/libpam-doc** и является частью пакета *libpam-doc*, который нужно устанавливать отдельно. Если вы найдете файл под названием **Linux-PAM System Administrators' Guide**, это отличное начало.
Если локальной копии нет, ее можно скачать на странице http://kernel.org/pub/linux/libs/pam/Linux-PAM-html/Linux-PAM_SAG.html.



Скорая помощь

Если вы что-то напутаете с настройкой PAM, то запросто можете захлопнуть дверь в свою квартиру, капитально запретив любой вход в систему, даже для администратора. Поэтому сохраните копию исходных файлов конфигурации pam, и будьте готовы выполнить экстренную загрузку, если понадобится их восстановить.

Часть 2: Экспериментируем с PAM

1 Позволим всем пользователям переключаться в root без пароля через su.

Что делать: Замените все строки файла /etc/pam.d/su, относящиеся к стеку auth, единственной строкой:

```
auth sufficient pam_permit.so
```

Как проверить: Зайдите в систему как обычный пользователь и проверьте, что можете переключиться в root с помощью su без ввода пароля. Это может показаться рисковым, но все же менее опасно, чем разрешить пользователям сразу заходить в систему с привилегиями администратора. Например, вы можете зайти в систему как обычный пользователь, а затем выполнить команду

```
$ su -c "date 09201155"
```

чтобы установить системное время (действие, требующее привилегий администратора).

2 Запретим всем пользователям использовать su

Что делать: Оставьте в файле /etc/pam.d/su только одну строку, относящуюся к стеку auth:

```
auth requisite pam_deny.so
```

Как проверить: Зайдите в систему как обычный пользователь и убедитесь, что переключения в режим администратора не происходит: у вас даже не спрашивают пароль. Конечно, можно запретить *su*, и сняв права на выполнение файла самой программы. Хотя отключение *su* на первый взгляд повышает безопасность системы, на деле вы вынуждаете администратора всегда заходить в систему как root, вместо кратковременного переключения в root для действий с высшими привилегиями.

3 Разрешим su только пользователям из некоторой группы

Что делать: Для начала нам нужен хотя бы один пользователь в этой группе – назовем ее, например, wheel. Чтобы добавить пользователя chris в группу wheel, выполните следующую команду от имени администратора:

```
# usermod -G wheel chris
```

Теперь отредактируйте файл **/etc/pam.d/su**, чтобы стек auth выглядел следующим образом:

```
auth sufficient pam_rootok.so
auth required pam_wheel use_uid
auth include system-auth
```

Как проверить: Зайдите в систему как chris (или как любой другой пользователь из группы wheel) и попробуйте использовать *su*. Будет запрошен пароль администратора, после чего должен открыться сеанс администратора. Теперь попробуйте зайти в систему под другим пользователем – не членом группы wheel, и сделать то же самое. Снова будет запрошен пароль администратора, но переключиться в сеанс root не удастся. Можете попробовать изменить управляющий флажок для модуля pam_wheel в requisite – для этого измените строку в файле **/etc/pam.d/su**:

```
auth requisite pam_wheel use_uid
```

Вопрос на засыпку: какое произойдет изменение? Проверьте себя. Ну что, начали понимать, как настройки PAM формируют политику безопасности?

4 Запретим вход в систему как root

Идея здесь в том, чтобы пользователи не могли войти в систему как администраторы: то есть зайти в систему придется как обычному пользователю, а затем переключиться в режим root с помощью *su*. (Предварительно убедитесь, что есть хотя бы один пользователь, имеющий на это право, не то вы заблокируете компьютер.) Нам нужно запретить вход как на виртуальные терминалы, так и на графический рабочий стол.

Что делать: Чтобы запретить вход на виртуальные терминалы, добавьте в файл **/etc/pam/d/login** строку

```
auth required pam_securetty.so
```

на верхушку стека auth (может быть, она уже есть там). Как мы уже видели, этот модуль запрещает вход в систему от имени root на терминалах, которых нет в списке **/etc/securetty**. Теперь мы можем очистить этот список с помощью пары команд:

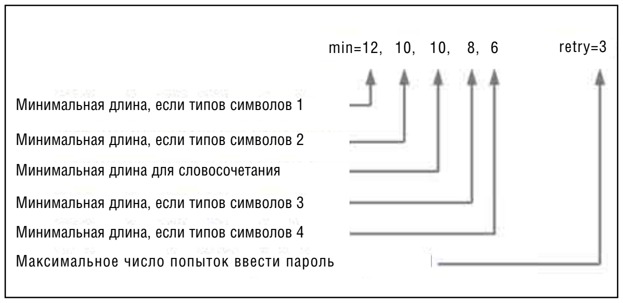
```
# cp /etc/securetty /etc/securetty.saved
# echo "" > /etc/securetty
```

(Обратите внимание, что пустой файл **/etc/securetty** и отсутствие этого файла – совсем не одно и то же. В первом случае запрещается вход под root на все терминалы, а во втором, наоборот, разрешается.) Далее, чтобы запретить вход от имени root в графический рабочий стол, добавьте ту же самую строку в стек auth в файле **/etc/pam.d/gdm**.

Как проверить: Попытайтесь зайти в систему как root (с правильным паролем) – и через виртуальный терминал, и через графический рабочий стол. Обе попытки должны потерпеть неудачу. Убедитесь, что вы можете зайти в систему как обычный пользователь. Обратите внимание, что если прибавить к этому **Эксперимент 3**, безопасность системы существенно повысится, потому что для получения доступа администратора придется сначала зайти в систему как пользователь из группы wheel, а затем открыть сеанс администратора через *su*. Даже пронюхав пароль администратора, взломщик не получит его привилегий.

5 Используем сложные пароли

В этом эксперименте мы используем модуль pam_passwdqc стека password, чтобы заставить пользователей назначать сложные пароли. (Это сработает только в том случае, когда пользователь задает или изменяет свой пароль, и не позволит выявить «слабости» уже действующих паролей.)



› **Рис. 3: Параметры модуля pam_passwdqc.**

Что делать: В Red Hat для проверки длины паролей уже используется модуль pam_cracklib, вызываемый из файла **system-auth**. Нам просто нужно заменить строку pam_cracklib следующей строкой:

Приемлем ли пароль?

Пароль	Длина	Типов символов	OK?
monpiastoria	12	1	Да
LinuxFormat	11	2	Да
beld*Grob	9	3	Да
4Me+You	7	4	Да
splodgerat	10	1	Нет
Four+five	9	2	Нет †
gosH!!2	7	3	Нет *
x4Z!	4	4	Нет

† (Начальные заглавные не в счет) * (Замыкающие цифры не в счет)

```
password requisite pam_passwdqc.so min=12,10,10,8,6
retry=3
```

Рис. 3 иллюстрирует параметры модуля pam_passwdqc. Качество пароля определяется его длиной, и в зависимости от количества типов символов, используемых в пароле, можно задать разные значения минимальной длину пароля. Существуют четыре типа символов: буквы нижнего и верхнего регистров, цифры и остальные символы. Заглавная буква в первом символе и цифра в последнем символе не в счет.

Как проверить: Попробуйте изменить свой пароль. Задавайте разные строки и в каждом случае подсчитывайте общее число символов и число типов символов, предполагая, что пароль должен быть приемлемым. Несколько примеров приведены во врезке «Допустим ли пароль?».

6 Запретим обычным пользователям выключать систему

Во многих системах обычным пользователям (не администраторам) дозволено выключать или перезагружать систему с помощью команд *halt*, *shutdown* и *reboot* – для рабочей машины предприятия, пожалуй, не лучшая идея. В данном эксперименте мы изменим конфигурацию PAM для этих команд, чтобы один лишь root мог выключать компьютер.

Что делать: Откройте файл настроек PAM для команды *halt* (**/etc/pam.d/halt**). В моем Red Hat стек auth для команды *halt* выглядит так:

```
auth sufficient pam_rootok.so
auth required pam_console.so
```

Измените его следующим образом:

```
auth sufficient pam_rootok.so
auth required pam_deny.so
```

Как проверить: Зайдите в систему как обычный пользователь и попробуйте остановить систему командой *halt*. У вас больше не должно этого получаться. Для полноты решения внесите те же изменения в файлы настроек для команд *shutdown* и *reboot*.

Д-р Крис Браун – автор книги *SUSE Linux: A Complete Guide to Novell's Community Distribution* (Издательство "O'Reilly", 2006). Некоторые главы можно найти на странице www.oreilly.com/catalog/suselinux/. LXF

Часто используемые модули PAM

Модуль	Действия	Описание
pam_unix	**auth, session, password**	Выполняет аутентификацию в традиционном стиле Unix, используя хэши паролей, которые хранятся в **/etc/shadow**. Включен в файлы настроек PAM многих приложений, непосредственно или с помощью директивы include.
pam_limits	**session**	Устанавливает ограничения на системные ресурсы, отпускаемые на время пользовательского сеанса. По умолчанию эти ограничения берутся из файла **/etc/security/limits.conf**; здесь можно задать «жесткие» и «мягкие» границы для таких параметров, как максимальное число процессов, максимальный размер файла, процессорное время и т.д.
pam_rootok	**auth**	Завершается успешно, если вы – администратор системы, и возвращает ошибку в противном случае. Обычно используется в сочетании с другими модулями аутентификации для реализации правила «Если вы администратор, можете продолжить и выполнить действие, если нет – необходима авторизация». Стек для этой политики может выглядеть так: `auth sufficient pam_rootok.so` `auth required pam_unix.so`
pam_cracklib	**password**	Проверяет качество пароля, затем еще раз проверяет пароль по системному словарю и ряду правил «плохого подбора пароля». Этот модуль обычно используется в стеке password, чтобы проверить качество пароля перед передачей его следующему модулю стека (обычно pam_unix), который обновляет пароль.
pam_passwdqc	**password**	Альтернативный модуль для проверки качества пароля, который поддерживает словосочетания (pass phrases) и может генерировать случайные пароли. Как и pam_cracklib, этот модуль обычно используется в стеке password для проверки качества пароля перед его обновлением. Подробный пример приведен в **эксперименте 5**.
pam_permit	**auth, account, session, password**	Всегда говорит «да» и возвращает успешный результат выполнения.
pam_deny	**auth, account, session, password**	Всегда говорит «нет» и возвращает ошибку. Обычно он используется только в самом низу стека для гарантированного статуса «ошибка».
pam_warn	**auth, account, session, password**	Просто отправляет сообщение (включая имя сервиса, терминал, имя пользователя и удаленный хост) службе журнала syslogd. Может быть использован, например, ближе к нижней части стека PAM для записи в журнал информации о неудавшихся попытках аутентификации как раз перед тем, как модуль pam_deny отклонит их.
pam_motd	**session**	Выводит «приветствие дня» (по умолчанию это файл **/etc/motd**), обычно после успешного входа в систему.
pam_securetty	**aurg**	Нет, это не опечатка в слове "security". Модуль pam_securetty разрешает вход в систему от имени root только с «безопасных терминалов» ("secure terminals"), перечисленных в файле **/etc/securetty**. (Запомните, tty на языке Linux означает «терминальное устройство»). Модуль никак не влияет на вход в систему для остальных пользователей. Этот способ управления доступом имел больше смысла в те дни, когда Unix работал на многопользовательских мини-компьютерах с 20 терминалами, включенными в порты RS-232; файл **securetty** мог использоваться для ограничения входа от имени root терминалом, запертым в офисе системного администратора. Однако он способен послужить и для полного запрета входа в систему под пользователем root... К этой идее мы еще вернемся в тексте.
pam_wheel	**auth, account**	Используется программами типа *su*. Этот модуль разрешает администраторский доступ только пользователям, включенным в группу wheel. (Историческая справка: Использование названия "wheel" для группы привилегированных пользователей уходит корнями в давние времена. Оно точно было в ранних версиях Unix, и пришло, если верить файлу жаргона (http://www.catb.org/jargon/index.html, из ОС Twenex, распространенной в шестидесятых годах. Но я понятия не имею, почему ее называют "wheel".)
pam_winbind	**auth**	Аутентифицирует пользователей в соответствии с учетными данными домена Windows путем взаимодействия с демоном *winbind* (который, в свою очередь, взаимодействует с контроллером домена Windows). Это важный компонент для ситуаций, когда вы добавляете Linux в существующую инфраструктуру на базе Windows и хотите поддерживать между этими системами общую базу данных учетных записей пользователей.
pam_nologin	**auth, account**	Запрещает пользователям (кроме root) вход в систему, если существует файл **/etc/nologin**. В некоторых дистрибутивах этот файл создается программой *shutdown*, чтобы запретить вход в систему во время ее остановки.

Файлы: ищите

Засунули что-то неизвестно куда? Рецепт майонеза? Письма тетушки Гвен? Огромный скачанный файл? Не беда! **Д-р Крис Браун** все вам найдет.



Наш эксперт

Д-р Крис Браун внештатный консультант по Linux со степенью доктора наук по физике элементарных частиц и обладатель сертификатов Novell CLP и Red Hat RHCE. Он только что написал книгу по SUSE Linux для O'Reilly.

Один из способов оценки относительной важности задач, ради которых люди используют Linux – определение количества различных приложений, написанных для решения этих задач. Исходя из огромного количества программ «поиска чего-нибудь», можно заключить, что чаще всего люди сразу же что-то теряют! На этом уроке мы проведем обзор ряда приложений, с помощью которых можно искать файлы и прочие данные по их названию, атрибутам или содержимому.

Поиск по имени файла

Самый простой поиск – это поиск по имени файла, и здесь удобно начать с шаблона имени файла в оболочке. Например, команда

```
$ ls *invoice*
```

выведет список всех файлов текущего каталога, имена которых содержат строку invoice. Не очень впечатляет? Тогда попробуйте

```
$ ls */*invoice*
```

Эта команда выведет список всех файлов со строкой invoice в имени, которые содержатся в текущем каталоге и во всех его подкаталогах. Можно расширить диапазон поиска до любого желаемого уровня, например, так:

```
$ ls *invoice* */*invoice* */*/*invoice*
```

Для поиска файла по имени во всей файловой системе подойдет утилита *slocate*. Например, команда

```
$ slocate invoice
```

найдет все файлы, имена которых содержат строку invoice. Так как *slocate* использует индексированный список имен файлов, то работает она с молниеносной быстротой. Этот индекс формируется программой updatedb (то же самое делает команда *slocate* с ключом *-u*), обычно запускаемой раз в день с помощью *cron* или *anacron*. В моем дистрибутиве Ubuntu 7.04 база данных *slocate* находится в файле **/var/lib/slocate/slocate.db**. Это единственный недостаток *slocate* – команда не найдет файлы, которые были созданы после последнего запуска *updatedb*.

S значит «безопасный»

Если вам интересно, то s в названии команды *slocate* восходит к слову «безопасный» ('secure'). И вот в чем здесь дело: программа *updatedb* (которая строит индекс) работает с привилегиями администратора, чтобы иметь доступ ко всем файлам. Поэтому в индексе **slocate.db** могут быть файлы, невидимые обычным пользователям (например, системные файлы или личные файлы других пользователей). Индекс *slocate* включает информацию о владельце файла и правах доступа, и программа *slocate* не покажет те файлы, соваться в которые вам не полагается. По-моему, раньше была еще программа *locate*, не отличавшаяся такой щепетильностью, но в современных дистрибутивах Linux *slocate* и *locate* указывают на одну и ту же утилиту.



› В этом примере добавлен еще один критерий поиска, но при желании можно задать их намного больше.

Спецпоиск: which и whereis

Для полноты картины упомянем о более узко направленных утилитах для поиска: *whereis* и *which*. Программа *whereis* производит поиск исполняемого файла, исходного кода и документации (страницы руководства) для заданной команды, просматривая каталоги из заранее определенного списка. Например, команда:

```
$ whereis ls
ls: /bin/ls /usr/share/man/man1/ls.1.gz
```

выводит информацию о местонахождении исполняемого (двоичного) файла и man-страницы для команды *ls*. Команда *which* еще более специализирована. Она просто ищет путь до заданной команды и выводит первый результат. Например, команда

```
$ which vi
/usr/bin/vi
```

сообщает, что команда *vi* – это исполняемый файл **/usr/bin/vi**. По сути, это ответ на вопрос: «Если бы я ввел команду *vi*, то какая программа запустилась бы на самом деле?"

Накачанный поиск: find

На другом конце шкалы – утилита-чемпион поиска, *find*. Кроме поиска по имени файла, через *find* возможен поиск по владельцу, правам доступа, времени последнего доступа, размеру файла и многим другим критериям. Плата за эту гибкость, естественно, сложный синтаксис командной строки. Вот вам пример, чтобы вы ухватили идею, а в детали мы углубимся позже:

```
$ find /etc -name '*.conf' -user cupsys -print
find: /etc/ssl/private: Permission denied
find: /etc/cups/ssl: Permission denied
/etc/cups/cupsd.conf
/etc/cups/printers.conf
```

В этом примере команда *find* ищет в каталоге **/etc** (и во вложенных каталогах) все файлы, имена которых заканчиваются на **.conf** и владельцем которых является пользователь cupsys.

Прочь, дурные вести...

В выводе *find* часто можно видеть вереницы сообщений об ошибках, касающихся каталогов, к которым у вас нет прав доступа. Иногда из-за них невозможно добраться до значимой информации. А подавить их легко: отправьте их в устройство «черной дыры» – **/dev/null**, приписав к командной строке 2> /dev/null.

и обрящете!

В общем случае синтаксис команды *find* таков:

```
$ find <где искать> <что искать> <что с этим делать>
```

Часть «где искать» – просто список каталогов для поиска, разделенных пробелами. Для каждого из них find рекурсивно спустится во все вложенные каталоги. В таблице *«Критерии поиска find»* на следующей странице перечислены самые полезные критерии поиска – часть «что искать», а на стр. 69 в таблице поменьше, *«Действия find»*, перечислены самые полезные действия (часть «что с этим делать»). Оба эти перечня не полны: для более подробной информации обратитесь к man-странице. Если никакого действия не задано, подразумевается -print, в результате которого путь и имя файла передаются в стандартный поток вывода – так команда *find* используется чаще всего. Пожалуй, стоит упомянуть, что многие критерии поиска *find* используются скорее с целью выполнения над найденными файлами каких-либо административных операций (допустим, резервного копирования), чем для того, чтобы помочь найти файлы, которые вы случайно посеяли.

Почему это не команда?

Иногда результат команды *which* может ввести в заблуждение, если переданная ей команда окажется также и встроенной командой оболочки *bash*. Например, результат выполнения команды

```
$ which kill
/bin/kill
```

сообщает, что утилита *kill* находится в каталоге **/bin**. Однако *kill* – это и встроенная команда *bash*, поэтому если я введу команду типа

```
$ kill -HUP 1246,
```

будет выполнена встроенная команда *kill*, а не внешняя.

Чтобы узнать, является ли команда встроенной, псевдонимом (alias) или внешней, можно использовать команду *type*, например, так:

```
$ type kill
kill is a shell builtin
```

Учимся на примерах

Чтобы разобраться во всем синтаксисе команды, потребуется время, поэтому, быть может, пригодятся некоторые примеры...

Пример 1 Это простой поиск по имени файла. Поиск начинается в моем домашнем каталоге, ищутся все файлы *PowerPoint* (**.ppt**). Обратите внимание, что мы поместили шаблон имени файла в кавычки, чтобы оболочка не развернула его. Мы хотим передать команде именно аргумент '***.ppt**', а о соответствии шаблону пусть заботится *find*.

```
$ find ~ -name '*.ppt'
```

Пример 2 В разделе «что искать» может быть несколько условий, и по умолчанию они объединяются через «логическое И», т.е. при поиске отбираются файлы, для которых выполняются все условия. К примеру, поищем подкаталоги в **/var**, владельцем которых является daemon:

```
$ find /var -type d -user daemon
```

Пример 3 Показывает, как объединить условия по «ИЛИ» вместо объединения по «И». В каталоге **/etc** мы ищем либо файлы, владельцем которых является cupsys, либо пустые файлы:

```
$ find /etc -user cupsys -or -size 0
```

Пример 4 Оператор ! используется, чтобы получить отрицание от условия. Найдем файлы в каталоге **/bin**, владельцем которых не является root:

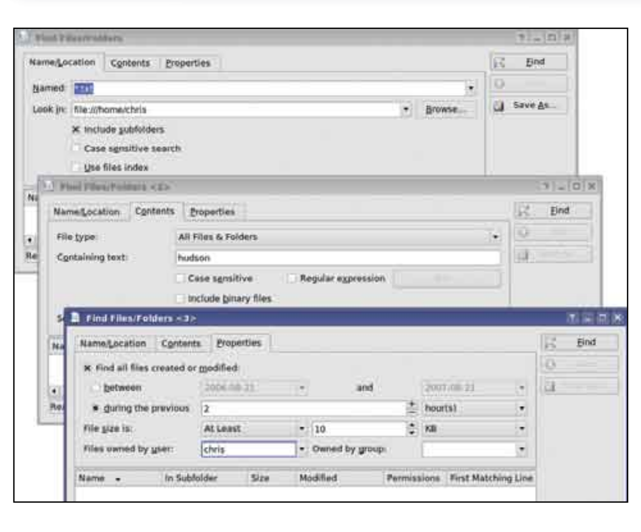
```
$ find /usr/bin ! -user root
```

Пример 5 Условия, содержащие числовые сравнения, часто сбивают с толку. Помните, что "+" перед числом означает «больше чем», "-" – «меньше чем», а если нет ни того, ни другого – *find* ищет точное соответствие. В трех строках ниже производится три поиска файлов: тех, что были изменены за последние десять минут, более чем год назад и ровно 4 дня назад (третий поиск, наверное, не самый остро необходимый).

```
$ find ~ -mmin -10
$ find ~ -mtime +365
$ find ~ -mtime 4
```

Пример 6 Из всех условий, наверное, сложнее всего те, в которых есть проверка по правам доступа к файлу. Вот неплохой пример – он производит поиск файлов с правами доступа 644 (в символьном виде их можно представить как rw-r--r--):

```
$ find ~ -perm 644
```



› **Мои эксперименты показали, что *KFind* не использует *find*.**

Пример 7 Поищем файлы, изменять которые разрешено каждому (т.е. и владельцу, и группе и всем остальным). Приведенные варианты одинаковы; в первом используется традиционная запись в восьмеричной системе, во втором – символьная запись, применяемая командой *chmod*:

```
$ find ~ -perm -222
$ find ~ -perm -ugo=w
```

Пример 8 Теперь ищем файлы, изменять которые можно кому-то конкретно: или владельцу, или группе владельца, или остальному миру

```
$ find ~ -perm /222
$ find ~ -perm /ugo=w
```

Пример 9 Пока мы использовали только действие по умолчанию -print, которое выводит список файлов. Вот пример, в котором используется действие -exec, которое перемещает все найденные файлы в каталог для резервного копирования. Здесь есть несколько вещей, которые нужно отметить. Обозначение {} будет заменено на полный путь найденного файла, а ';' используется для обозначения окончания команды, следующей за -exec. Помните, что ';' – это также специальный символ оболочки, и нужно поставить перед ним обратный слэш, чтобы оболочка его не обрабатывала. »

Скорая помощь

В Linux доступен поиск через *Google Desktop 1.0*. Для запуска он требует *glibc 2.3.2+* и *gtk+ 2.2.0* и выше. В нашем обзоре в LXF97 мы дали ему 6/10, так как сочли, что ему еще есть куда расти. Загрузите его и сделайте выводы сами!

Критерии поиска для *find*

СИНТАКСИС	ОПИСАНИЕ	ПРИМЕР
-name string	Имя файла соответствует строке (можно употреблять шаблоны)	**-name '*.jpg'**
-iname string	То же самое, что -name, но без учета регистра	**-iname '*tax*'**
-user username	Владельцем файла является username	**-user chris**
-group groupname	Группой владельца файла является groupname	**-group admin**
-type x	Файл типа 'x'. Возможные типы: f – обыкновенный файл d – каталог l – символическая ссылка c – символьное устройство b – блочное устройство p – именованный канал (FIFO)	**-type d**
-size +N	Размер файла больше N блоков по 512 байт (суффикс c – для байт, k – для килобайт, M – для мегабайт)	**-size +100M**
-size -N	Размер файла меньше N блоков (суффикс c – для байт, k – для килобайт, M – для мегабайт)	**-size -50c**
-mtime -N	Последнее изменение файла было менее чем N дней назад	**-mtime -1**
-mtime +N	Последнее изменение файла было более чем N дней назад	**-mtime +14**
-mmin -N	Последнее изменение файла было менее чем N минут назад	**-mmin -10**
-perm mode	Точное соответствие прав доступа к файлам. Права доступа могут быть записаны в восьмеричном виде или в символьной нотации, поддерживаемой chmod	**-perm 644**
-perm -mode	Установлены все биты разрешений, указанные в mode	**-perm -ugo=x**
-perm /mode	Установлен любой из битов разрешений, указанных в mode	**-perm /011**

```
$ find ~ -mtime +365 -exec mv {} /tmp/mybackup \;
```

Неважно, как его звать; внутри-то что?

Как мы убедились, средства поиска типа *find* могут искать файлы по имени, размеру, владельцу, времени доступа и многим другим параметрам, но не могут искать файлы по содержимому. Оказывается, файлы достаточно изящно находятся по содержимому при помощи *grep*, используемой совместно с шаблонами оболочки для файлов. Вот пример, взятый из моей собственной файловой системы:

```
$ grep -l Hudson */*
Desktop/suse_book_press_release.txt
google-earth/README.linux
Mail/inbox.ev-summary
Mail/sent-mail.ev-summary
snmp_training/enterprise_mib_list
```

Здесь мы просим *grep* вывести имена файлов, содержащих строку Hudson. Шаблон */* разворачивается оболочкой и означает список всех файлов, находящихся на один уровень ниже текущего каталога. Если нужно наложить некоторые условия на имя файла, это можно сделать примерно так:

```
$ grep -l Hudson */*.txt
Desktop/search_tools.txt
Desktop/suse_book_press_release.txt
```

тогда команда будет искать только файлы с расширением **.txt**. В принципе, можно расширить диапазон поиска и на большее количество уровней, но на практике весьма вероятна ситуация, когда подставленных файлов будет столько, что строка со списком аргументов превысит допустимую длину. С этим я и столкнулся, попробовав

```
$ grep -l Hudson */* */*/*
bash: /bin/grep: Argument list too long
```

Более эффективный подход – использовать *grep* совместно с *find*. Поиск файлов с расширением **.txt**, содержащих строку Hudson и находящихся в моем домашнем каталоге ('**~**'), выглядит так:

```
$ find ~ -name '*.txt' -exec grep -q Hudson {} \; -print
/home/chris/Desktop/search_tools.txt
/home/chris/Desktop/suse_book_press_release.txt
```

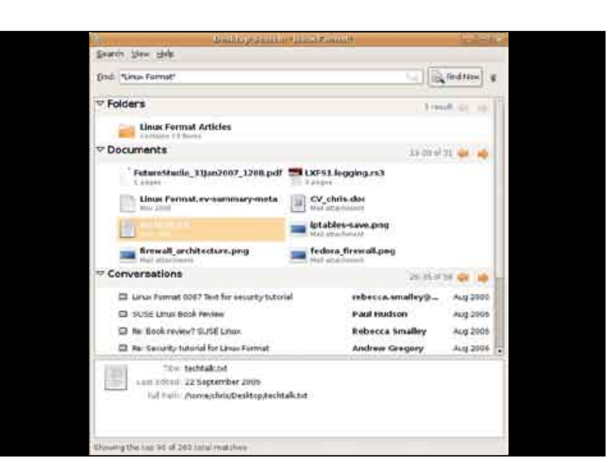
Правда о *find*

Отдельные части синтаксиса команды *find* называют выражениями (или, более формально, предикатами). Например, -uname cupsys – это предикат. Команда *find* проверяет каждый файл в заданном каталоге и вычисляет значения всех предикатов для этого файла. Каждый предикат возвращает **true** или **false**, и результаты всех предикатов объединяются логическим «И». Если хотя бы один предикат возвращает значение **false**, значения остальных не вычисляются.

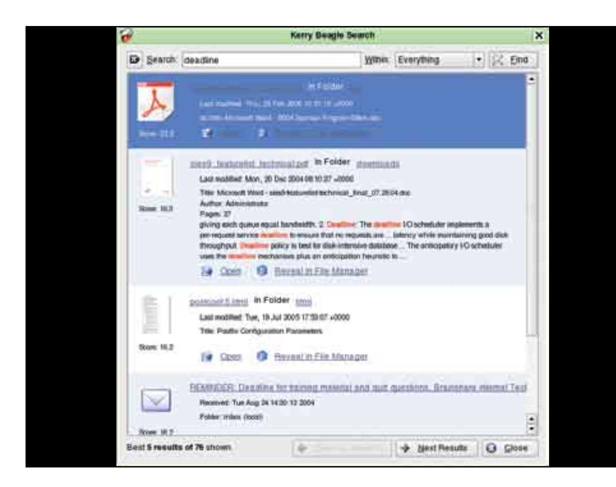
Рассмотрим команду:

```
$ find . -user chris -name '*.txt' -print
```

Если предикат -user chris возвращает значение **false** (т.е. chris не является владельцем файла), то значения остальных предикатов не вычисляются. Только если каждый из предикатов -user chris и -name '*.txt' возвращает **true**, вычисляется предикат –print (который выводит имя файла в стандартный поток вывода и тоже возвращает **true**).



› **Ищем 'Linux Format' на рабочем столе Gnome.**



› Жителям графства Керри в Ирландии, наверно, приятно, что оно ассоциируется и с таким умником, для разнообразия...

Этот подход ликвидирует проблему переполнения списка аргументов, которая повредила нам в предыдущем случае. Помните, что *find* может искать файлы не только по имени, но и по многим другим параметрам, а *grep* позволяет использовать регулярные выражения, а не только фиксированный текст, поэтому возможности здесь гораздо шире, чем может показаться из этого простого примера. Если его синтаксис непонятен, прочтите врезку «Правда о find» слева вверху. В этом примере предикат -exec grep -q Hudson {} \; возвращает **true**, если grep находит вхождение строки Hudson в заданном файле, и **false**, если нет. Если значение предиката – **false**, *find* не вычисляет следующие выражения, т.е. не выполняет действие -print.

Графические средства

До сих пор мы занимались утилитами поиска, работающими в командной строке; но, конечно, имеются и графические средства. В Gnome есть графический инструмент поиска *gnome-search-tool*, показанный на стр. 68. После запуска программа предоставляет минимально необходимый интерфейс, через который указывается маска для имени файла и каталог для поиска. Постепенно можно добавить новые критерии поиска; некоторые из них показаны на рисунке. Эти критерии понятны из нашего разговора о *find*, и, честно говоря, для поиска *gnome-search-tool* в фоновом режиме запускает *find*. Откуда я это знаю? Ну, мы попробовали переименовать исполняемый файл *find* и обнаружили, что после этого *gnome-search-tool* выдал ошибку «Не могу запустить дочерний процесс 'find'».

В KDE есть похожее средство под названием *KFind*, с немного иначе организованным интерфейсом: критерии поиска разделены на три вкладки, показанные на стр. 67.

Пустим ищейку по следу

Названный по имени ищейки, известной острым нюхом и умением выслеживать, *Beagle* – в другой лиге средств поиска. Цитирую страницу проекта *Beagle* (http://beagleproject.org): «*Beagle* – это средство поиска, которое исследует ваше личное информационное пространство и находит все, что бы вы ни искали». Оно может искать в документах самого различного типа: обычном тексте, документах *OpenOffice.org* и *Microsoft Office*, файлах PDF и HTML, man-страницах, в других источниках информации, таких как почтовые папки и адресные книги *Evolution* и *KMail*, заметки в *Tomboy* и *KNotes* и RSS-листы. (Полный список можно найти на странице http://beagle-project.org/Supported_Filetypes).

Beagle написан на .NET, и ему необходимы среда выполнения Mono и несколько библиотек. Полтора года назад я писал книгу, и, пытаясь получить для нее рабочую версию *Beagle*, столкнулся с мириадами зависимостей и проблемой несовместимости версий. Сейчас, похоже, программа созрела – с большинством современных дистрибутивом поставляется *Beagle*, который работает «из коробки», а иногда *Beagle* интегрируется в Gnome. Внизу показан снимок *Beagle* в Ubuntu 7.04. Я поискал по фразе "Linux Format" и нашел немало вхождений в файловой системе и в почтовом архиве. Для рабочего стола KDE есть графический клиент *Kerry Beagle*; на экранном снимке в верхней части страницы он показан запущенным в SUSE Linux. Если вам интересно, *Kerry Beagle* – другая разновидность гончих; то, что ее название начинается с 'K' – сущий подарок для фанатов KDE.

Для быстрого поиска *Beagle* использует предварительно сформированный индекс, но этот индекс гораздо динамичнее того, что раз в день создает программа *slocate*. При первом запуске, *Beagle* забирается в ваш домашний каталог и индексирует все данные. Если у вас много файлов, сообщений электронной почты или других документов, или ваш компьютер сильно загружен, для полного индексирования всех данных может потребоваться несколько часов. *Beagle* также использует *inotify* – индекс динамически обновляется, если в системе происходят какие-то изменения. Индексирование выполняется демоном *beagled*, который запускается под обычным пользователем (не под администратором) и имеет доступ только к вашему домашнему каталогу. *Beagle* – отличное средство поиска информации в вашем личном пространстве, но не в системных файлах. Процесс индексации специально ограничивает использование ресурсов процессора во избежание чрезмерной загрузки компьютера. Однако знайте, что индексы съедают много дискового пространства. FAQ по *Beagle* предупреждает, что размер индекса составляет 5–10 процентов от размера индексируемых данных, но в моей системе он занял около двух процентов (размер индекса – 71 МБ, а файловой системы – 3,6 ГБ). Индекс в виде иерархического набора файлов хранится в каталоге **~/.beagle**.

Хотя *Beagle* чаще всего используется через графический интерфейс, в нем есть и инструменты командной строки – в частности, *beagle-query*, и удобные средства администрирования, включая *beagle-config*, с помощью которого можно настроить процесс индексирования, и *beagle-status*, который предоставляет регулярно обновляемую информацию о ходе индексирования демоном *beagled*. LXF

Действия *find*

ДЕЙСТВИЕ	ОПИСАНИЕ
-print	Выводит полный путь с именем файла в стандартный вывод
-ls	Выводит полный листинг файла (эквивалентно команде **ls -dils**)
-delete	Удаляет файл
-exec command	Выполняет указанную команду. Все последующие аргументы до появления **;** считаются аргументами команды. Строка **{}** заменяется именем текущего файла.

ПК против людей

Хотя такие программы, как *find* и *Beagle*, впечатляют, нам пока еще далеко до машинных средств поиска, имитирующих возможности человека. Например, мы не можем спросить компьютер: «Где та картинка с коровами на пляже?» (если, конечно, не позаботились назвать ее **cows_on_beach.jpg**). Нельзя также потребовать найти все MP3-файлы с соло на виолончели. Поэтому – эй, молодежь! Где ваши таланты? Пора кодировать!

Другой софт

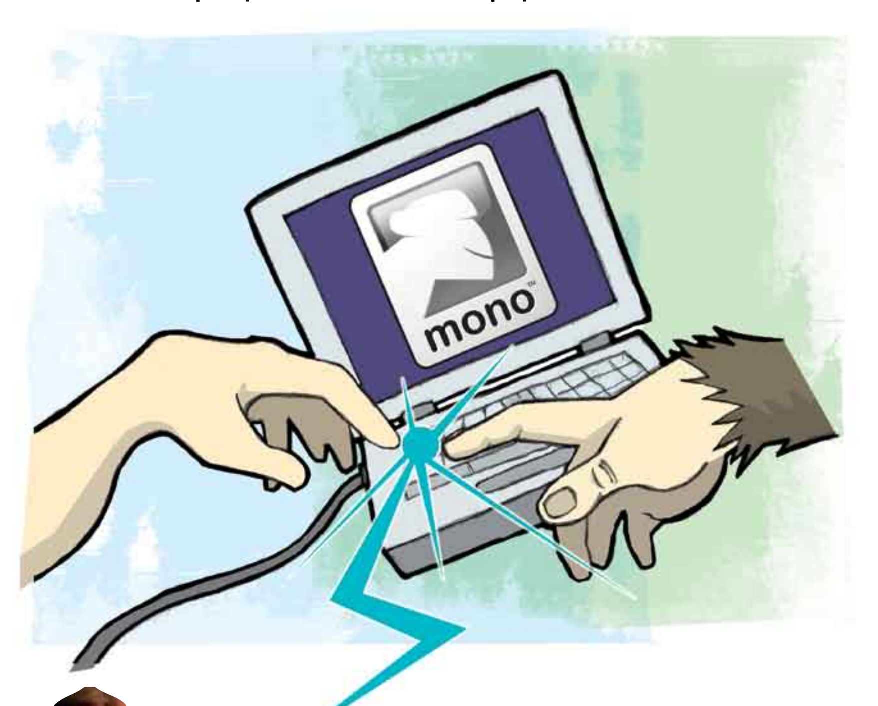
Searchmonkey – средство для сравнения файлов и их содержимого на *Gtk+*. http://searchmonkey.sourceforge.net

Strigi – небольшой поисковый демон для извлечения данных из файлов, например, длительности аудиороликов, содержимого документов или разрешения изображений. Не привязан к конкретному рабочему столу (Gnome или KDE). http://strigi.sourceforge.net

Tracker позволяет искать документы так же, как *Spotlight* в OS X. www.gnome.org/projects/tracker

Mono: Рецепты

За прошедший год **Пол Хадсон** рассказал нам о файловых системах, доступе к базе данных, графическом интерфейсе, XML и сетях. Пришло время расправить крылья...



Наш эксперт

Пол Хадсон полагает, что Mono – лучшая вещь со времен мультфильма Pinky and the Brain, и сейчас поддерживает два проекта на основе Mono на SourceForge.

Дошло, что от вас требуется? Чтобы стать хорошим программистом, необходимо нечто большее, чем пристрастие к сандалиям и очки, держащиеся только на клейкой ленте. На самом деле, это острый ум, хорошая память, жадность до новых технологий и – бесспорно, самое важное – способность хранить часто используемые участки кода в голове, чтобы решать стандартные проблемы быстро. Многое тут вытекает из склада ума, но ум приходит с опытом: вы сталкиваетесь с проблемой, решаете ее и создаете личную библиотеку кода, позволяющую быстро сколачивать решения. Данный урок – последний в нашей серии учебников по Mono, поэтому мы сосредоточимся на кусках кода, способных помочь вам решить стандартные проблемы и извлечь преимущества из полезных технических приемов – пожалуйста, берите их и свободно используйте в своих проектах, под какой бы лицензией они не издавались.

Проблема

Удалить в цикле несколько элементов из массива

Этот пример достаточно прост для затравки, но он ставит в тупик некоторых начинающих. Допустим, у вас есть объект List<string> с именем MyNames, то есть он хранит массив строк. Поскольку это обобщенный тип данных, вам следует добавить using System.Collections.Generic; в начало вашего файла проекта. И если вы хотите удалить из этого списка все имена, начинающиеся с "Mike", то первый вариант вашего кода может выглядеть так:

```
for (int i = 0; i < MyNames.Count; ++i) {
  if (MyNames[i].StartsWith("Mike")) {
    MyNames.RemoveAt(i);
  }
}
```

Или, если вы аккуратист, то так:

```
foreach (string name in MyNames) {
  if (name.StartsWith("Mike")) {
    MyNames.Remove(name);
  }
}
```

Но здесь налицо серьезная проблема: .NET не разрешает изменять массив, пока вы перемещаетесь по нему, то есть первый Mike будет удален, но цикл продолжится, и появится ошибка, потому что на самом деле вы сдвинули все элементы на одну позицию, и какой же элемент должен быть следующим в цикле? Приведенное решение вполне очевидно, если немного подумать: при перемещении по массиву в обратную сторону сдвиг элементов не имеет значения, потому что вы его уже обработали. Вот оно:

```
for (int i = MyNames.Count – 1; i >= 0; --i) {
  if (MyNames[i].StartsWith("Mike")) {
    MyNames.RemoveAt(i);
  }
}
```

Проблема

Округление чисел портит ваш код

По мере увеличения объема кода нарастает необходимость его чистки. Одной из наиболее раздражающих ловушек в коде на C# является Math.Round(), потому что если вы захотите написать код

```
int foo = Math.Round(10.1f);
```

он не сработает. О нет – вы получите сообщение об ошибке преобразования: Mono не умеет преобразовывать из double в int. Вы-то думали, что этот код преобразует число с плавающей точкой 10.1 в целое 10, но Math.Round() возвращает не целое – потому что если указать второй параметр, можно получить число, округленное до указанного знака после запятой.

Конечно, это лишнее, если вам всего лишь надо преобразовать число с плавающей точкой в целое, поэтому я предлагаю создать такой небольшой метод:

```
public int Round(float num) {
  return (int)Math.Round(num);
}
```

Вы можете использовать его так:

```
int foo = Round(10.1f);
```

Вам, вероятно, кажется, что можно и без него обойтись, но представьте такой код:

```
DrawRectangle((int)Math.Round(obj.x), (int)Math.Round(obj.y), (int)Math.
Round(obj.w),  (int)Math.Round(obj.h))
```

Ну не уродство? Странно: в Java есть прекрасный метод Math.round(), получающий float, а возвращающий int, а вот в C# требуется собственный код. Не опасайтесь снижения производительности за счет

» **Месяц назад** Назад, в Unix: мы заставили .NET прекрасно работать с Linux.

добавочного вызова функции: такой простой метод, вероятно, будет встроенным (inline).

Проблема

Сортировка массива экзотических данных

Стандартный класс List имеет метод Sort(), который выстраивает строки и числа в определенном порядке, но он бесполезен, если вы храните объекты и хотите отсортировать их по определенному свойству. Однако вы можете сообщить Sort() имя сравнивающей функции, способной выполнять более продвинутую сортировку, а затем использовать ее обычным способом. Например, пусть у вас есть класс

```
public class User {
  public int ID;
  public string Name;
}
```

и List [Список] этих пользователей, вроде такого:

```
List<User> MyUsers = new List<User>();

User user = new User();
user.ID = 1;
user.Name = "Paul";
MyUsers.Add(user);

user = new User();
user.ID = 10;
user.Name = "Scott";
MyUsers.Add(user);

user = new User();
user.ID = 5;
user.Name = "Mike";
MyUsers.Add(user);

user = new User();
user.ID = 50;
user.Name = "Graham";
MyUsers.Add(user);
```

Сортируя его при помощи обычного старого Sort(), вы получите ошибку, ибо .NET не умеет обращаться с объектами User. Но не так уж трудно написать собственный метод сравнения сложных объектов. А если вы предоставите его имя функции Sort(), он будет вызываться для каждого сравнения двух объектов, чтобы решить, в каком порядке их расположить. Метод должен возвращать **1** (объект 1 должен следовать после объекта 2), **-1** (объект 2 должен следовать за объектом 1) или **0** (объекты 1 и 2 равноправны). Он может выглядеть примерно так:

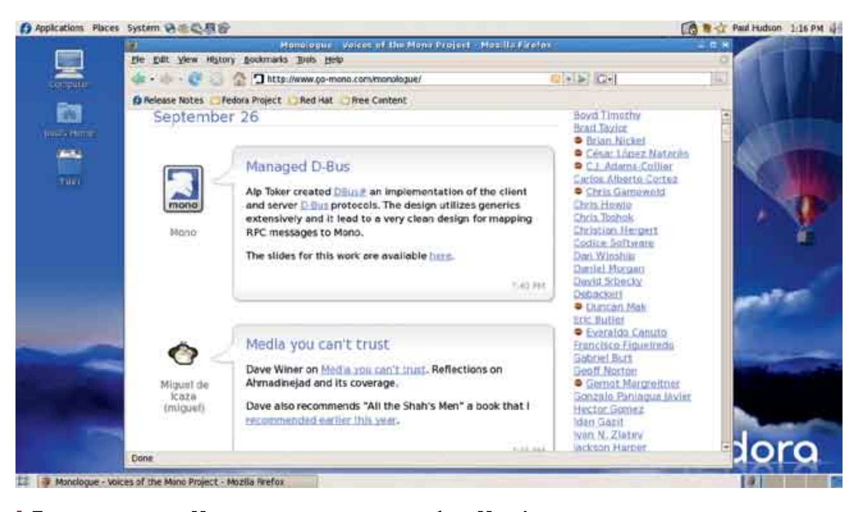
```
private int CompareUserByID(User a, User b) {
  if (a.ID > b.ID) {
     return 1;
  } else if (a.ID < b.ID) {
     return -1;
  } else {
     return 0;
  }
}
```

Затем сортируйте ваш массив, используя

```
MyUsers.Sort(CompareUserByID);
```

Теперь все элементы будут переставлены. Приведенный способ показывает, как создать свою собственную систему сортировки для любого типа данных, но встроенные типы данных – int, string и т.п. – можно сравнивать еще проще. Все эти типы имеют специальный метод CompareTo(), принимающий в качестве единственного параметра другой идентичный тип и возвращающий вам **1**, **-1** или **0**. Поэтому, если хотите, можете написать метод CompareUserByName() вот так:

```
private int CompareUserByName(User a, User b) {
  return a.Name.CompareTo(b.Name);
}
```



› Получите новости Mono прямо из первых рук: блог Monologue на www.go-mono.com/monologue содержит комментарии ведущих разработчиков Mono.

Проблема

Перемешать элементы массива случайным образом

.NET имеет несколько способов манипуляции массивами, но ни один из них не столь сжат, как функция shuffle() в PHP: даете ей массив и получаете обратно перемешанный. Вы можете скопировать ее одним из двух способов – в зависимости от того, хотите ли вы потренироваться или выглядеть круче!

Простой путь перемешать массив таков:

```
public void ShuffleList(List<string> list) {
  Random rand = new Random();

  for (int i = 0; i < list.Count; i++) {
    string tmp = list[i];
    list.RemoveAt(i);
    list.Insert(rand.Next(0, list.Count), tmp);
  }
}
```

В цикле перемещаемся по массиву, удаляя каждый элемент и вставляя его в случайную позицию. Заметьте: код генерирует случайное число, используя новый объект Random при каждом вызове метода; это лучше, чем создание одного объекта Random для всей программы.

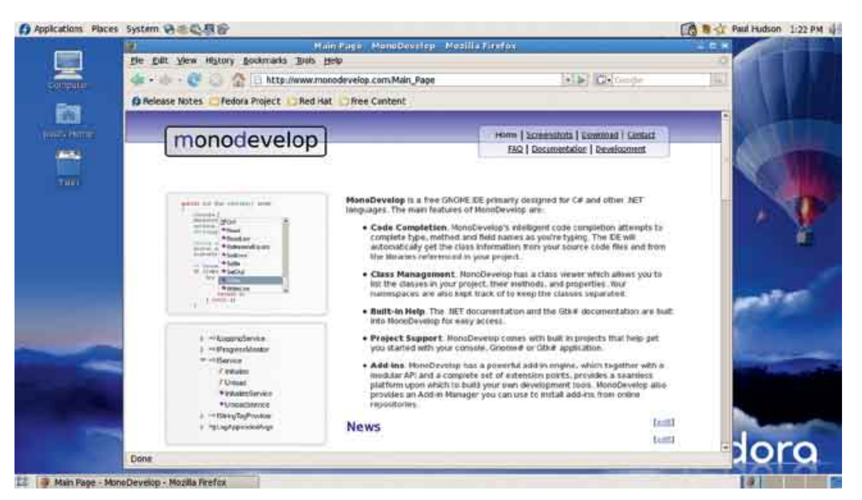
Итак, вот схема перемешивания: взять List, содержащий строки, и перемешать их случайным образом. А если вы захотите перемешать массив целых чисел? Или объектов User? Или чего угодно, но не строк? Можно, конечно, создать несколько методов ShuffleList(), но это недальновидное решение: ваш код очень скоро раздуется.

Намного лучше использовать стандарты, создав функцию, которая принимает список любого типа и перемешивает содержимое. Тут требуется некий специальный синтаксис C#, потому что вам необходимо сообщить своему методу, что он будет принимать неизвестный тип и использовать этот тип для всех данных. Обычно на обобщенные типы ссылаются как на T или T1, T2 и т.д., если их более одного. Итак, метод ShuffleList() можно переписать так:

```
public void ShuffleList<T>(List<T> list) {
  Random rand = new Random();

  for (int i = 0; i < list.Count; i++) {
    T tmp = list[i];
    list.RemoveAt(i);
    list.Insert(rand.Next(0, list.Count), tmp);
  }
}
```

Когда вы используете ShuffleList(MyUsers), .NET по сути заменяет в этом методе «T» на «User», т.е. ShuffleList() принимает List<User>,



› MonoDevelop.com – домашняя страница IDE, мы использовали ее во всех уроках; почаще справляйтесь об обновлениях, ведь они облегчают жизнь!

» а переменная tmp получает тип User. Итак, вы можете вызвать ShuffleList() со списком [List] строк, целых чисел, дробных, логических, людей, рыбок или данных любого другого типа, какой сможете придумать.

Проблема

Узнать, когда один объект находится над другим

Это весьма общая формулировка, а вот конкретный пример: вы хотите знать, когда мышь находится над нарисованным вами объектом. Проблема решается очень просто: все, что надо сделать – это проверить, что координаты мыши больше, чем позиция X и Y вашего объекта, и меньше чем X, Y + ширина и высота объекта. Вчерне можно записать подобный метод так:

```
public bool PointOverRect(int x1, int y1, int x2, int y2, int width, int
height) {
  if (x1 >= x2 && x1 <= x2 + width) {
    if (y1 >= y2 && y1 <= y2 + height) {
      return true;
    }
  }

  return false;
}
```

Для использования этого метода передайте координаты X и Y мыши в качестве первых двух параметров, затем X, Y, ширину и высоту вашего объекта в качестве вторых параметров. Конечно, реально это работает только для прямоугольных объектов, но создавайте прямоугольные рамки вокруг объектов другой формы, и все будет хорошо. [для фигур произвольной формы часто в качестве второго входного параметра используется массив координат узлов контура, ограничивающего объект или, если контур является геометрической фигурой, то ее атрибуты, например, центр и радиус окружности, – *прим. пер.*]

В принципе, можно взять этот простой метод и подстроить под свои личные цели. Например, вы могли бы захотеть добавить поддержку вращения ограничивающей рамки. Если все, что вам надо, это стандартная функция проверки нахождения точки в прямоугольнике, используйте метод Contains() вашего прямоугольника и передайте ему точку.

Проблема

Нужно узнать, перекрываются ли два объекта

Еще одна общая проблема, так что снова поясню на примере: вы хотите реализовать проверку столкновений в игре. Это очень похоже на проверку, принадлежит ли точка прямоугольнику, особенно если использовать метод Contains(). Однако, хотя Contains() и может принимать объект Rectangle в качестве параметра, он возвращает **true**, если один прямоугольник полностью лежит внутри другого, а не просто пересекает его, а для обнаружения столкновений вам необходимо последнее.

К счастью, для прямоугольников есть другой небольшой полезный метод, под названием Intersect(), который накладывает один прямоугольник на другой и возвращает новый прямоугольник-пересечение, и вы можете проверить его ширину и высоту, чтобы понять, имеет ли место пересечение. Простой и легкий способ проверки столкновений – вот такой метод:

```
public bool RectOverRect(int x1, int y1, int width1, int height1, int x2, int
y2, int width2, int height2) {
  Rectangle rectthis = new Rectangle(x1, y1, width1, height1);
  Rectangle rectthat = new Rectangle(x2, y2, width2, height2);

  rectthis.Intersect(rectthat);

  if (rectthis.Width == 0 && rectthis.Height == 0) {
    return false;
  } else {
    return true;
  }
}
```

Проблема

Обработка ошибок при их возникновении

Я не затрагивал старый добрый блок try/catch в нашей серии, но теперь настало время это сделать! Система try/catch позволяет выполнять команды и предпринимать заданные действия, если возникла ошибка. Например, вы можете написать:

```
try {
        SomeDangerousMethodCall();
} catch (Exception e) {
        Console.WriteLine("Ой!");
}
```

Обычно ошибка в SomeDangerousMethod() приводит к краху программы, но использование try/catch означает, что такая ошибка в SomeDangerousMethod() вернет управление в вызывающий код, с последующей передачей блоку catch, а тот выведет «Ой!», элегантно обработав вашу ошибку. Это не повод становиться программистом-неряхой, потому что код обработки исключений вроде этого здорово тормозит – уж лучше заранее выполнять проверки в коде!

Вы можете перехватывать несколько типов исключительных ситуаций, добавив новые блоки catch; выполнится лишь один, соответствующий конкретному исключению; а если возможны непредвиденные ситуации, следует, вероятно, добавить общий обработчик Exception – это базовый класс всех исключительных ситуаций, соответствующий всем исключениям вообще.

```
try {
  DangerousMethod();
} catch (DllNotFoundException e) {
  Console.WriteLine("Ой - отсутствует необходимая DLL!");
} catch (FileNotFoundException e) {
  Console.WriteLine("Ой - отсутствует необходимый файл!");
} catch (Exception e) {
  Console.WriteLine("Ой - произошла ошибка!");
}
```

Преимущество соответствия конкретному исключению в том, что вы получаете дополнительные данные для обработки. Например, FileNotFoundException имеет свойство FileName, которое подскажет, какой файл отсутствует.

В истинно устойчивой как скала программе следует использовать блоки try/catch почаще – при желании их даже можно вкладывать друг в друга, чтобы предусмотреть самые причудливые ошибки. Предусмотрены исключения для всех сортов типичных проблем:

OutOfMemoryExceptions, AccessViolationException и немаловажное NullReferenceException. Не скупитесь на проверки!

И, наконец, finally...

Было бы неверно описать try/catch, не сказав об его кузене try/finally. Он используется намного реже, чем try/catch, вследствие общего заблуждения, что в .NET-коде незачем беспокоиться об управлении памятью. Что ж, вот вам изящный поворот: всякий раз, когда вы беретесь за собственный код, будьте с памятью поосторожнее. Объекты, классы и ресурсы .NET все под контролем, то есть автоматически освобождаются, когда больше не нужны, но собственные ресурсы – например, 3D-текстуры, загруженные вами в OpenGL – не управляемы, и о них необходимо позаботиться вам. Блок try/finally разработан, чтобы обезопасить управление памятью, путем насильственного выполнения заданного блока кода, невзирая ни на что. Это полезно даже помимо управления памятью, потому что вы будете уверены, что определенный метод вызовется перед тем, как объект будет освобожден.

Перейдем от теории к практике на примере: когда клиент подключается к вашему серверу, вы хотите отослать ему текст приветствия, прочитать какой-то текст, отправить текст-прощание, затем закрыть сокет. Посмотрите этот код:

```
function ClientConnect(MySocket sock) {
        try {
                sock.SendHello();
                sock.ReadMessage();
                sock.SendGoodbye();
        } catch (Exception e) {
                Console.WriteLine("При подключении клиента возникла
ошибка.");
        }
}
```

Вам этот код может показаться вполне пригодным, но вдруг в процессе чтения возникнет ошибка – предположим, клиент отправит неправильно оформленное сообщение? Вот что произойдет:

1 Вызов SendHello()
2 Вызов ReadMessage()
3 Возникло исключение
4 Вызов Console.WriteLine()
5 Метод завершился

Как видите, в этой последовательности событий нет SendGoodbye(), то есть сообщение-прощание никогда не будет отослано. Если клиент ожидает его, или сервер использует этот метод для выполнения некой очистки собственных ресурсов, тогда у вас проблема. Тут-то и выходит на сцену try/finally, потому что он в общем гарантирует, что определенный блок кода вызовется при любом раскладе. Перепишем предыдущий пример:

```
void ClientConnect(MySocket sock) {
        try {
                try {
                        sock.SendHello();
                        sock.ReadMessage();
                } catch (Exception e) {
                        Console.WriteLine ("При подключении
клиента возникла ошибка.");
                }
        } finally {
                sock.SendGoodbye();
        }
}
```

Даже если в SendHello() или в ReadMessage() возникнет исключение, SendGoodbye все равно будет вызван. На самом деле, работает даже нечто вроде этого:

```
void ClientConnect(MySocket sock) {
        try {
                sock.SendHello();
                sock.ReadMessage();
                return;
        } finally {
                sock.SendGoodbye();
        }
}
```

Вызов return должен бы привести к немедленному выходу из метода, да и приводит – но .NET все-таки сначала выполняет все блоки finally. Даже старый метод Environment.Exit() находит время для вызова блоков finally перед завершением программы – а если вы не хотите, чтобы ваш блок finally выполнился (поэтому я и сказал, что блоки finally «в общем гарантируют», а не «абсолютно гарантируют» выполнение блока кода), используйте метод Environment.FailFast(). LXF

Идем дальше

Ну-с, Mono-маны, это конец: надеюсь, вы попытались завершить хотя бы один из наших проектов, потому что лучший способ обучения – это пытаться делать что-то самому, менять код под ваши нужды, создавать библиотеки идей и кода для выполнения ваших заказов. Берите код и технические приемы, описанные выше, и применяйте их в вашей работе – да сообщите мне, как ваши дела!

Перехватываемые исключения

AccessViolationException	Возникает, когда вы пытаетесь записать в область памяти только для чтения.
ArgumentNullException	Возникает, когда метод требует аргументы, а вы случайно передаете ему null.
DivideByZeroException	Деление любых чисел на ноль – табу в любом языке программирования; перехватывается здесь!
DllNotFoundException	Когда .NET создает ссылки на несуществующие родные библиотеки, возникает это исключение.
Exception	Дедушка всех исключений; хорош для перехвата, когда вы не представляете, что может произойти.
IndexOutOfRangeException	Возникает при выходе за границы и попытке чтения несуществующего элемента массива.
NullReferenceException	Вы получаете это при попытке читать из несозданного объекта.
OutOfMemoryException	Системе не хватает памяти, и, вероятно, ваша программа будет закрыта.

Ядро: Заделай

Заплатки могут привести к революции в системе, повысив эффективность взаимодействия ее частей. Полковник **Ник Вейч** отдает приказ о выдвижении…



Разработчики ядра – очень талантливые и замечательные люди (это они просили нас так сказать), так почему же вы недовольны кристальной чистотой типового ядра, выходящего из недр штамповочной машины на www.kernel.org? Прямо и не знаю... пойдем дальше, тут разбирать нечего... Ой. Погодите минутку. Ведь в жизни очень мало машин с Linux, где запущено чистое типовое ядро. Причина? Все популярные дистрибутивы латают ядра сами.

Они так поступают не потому, что им нечего делать, а потому что есть функции, внедрения которых они просто не могут ждать (или ошибки, оставлять которые нельзя). Многие из этих заплаток сконцентрированы вокруг проблем с производительностью, драйверами и файловыми системами, а многие внедряются в дистрибутивы собственными хакерами ядра, чей код не приняли в основную ветку разработки.

Идя путем сугубо личной настройки, можно добраться до самых разных мест. Например, при попытке загрузить ядро для Mandriva вам предоставят выбор из 55 (!) вариантов. Некоторые из них отличаются разными заплатками (например, специально для ноутбуков или для меньшей латентности настольных приложений), а некоторые просто имеют разные конфигурации ядра. Какие бы причины ни побудили создателей вашего дистрибутива добавить заплатки, безусловно, это дает достаточно оснований для вас сделать то же самое!



Наш эксперт

Ник Вейч
Добродушный олигарх Linux Format очень организован – в электронном смысле. Но бардак в его сарае такой же, как и в вашем: велосипеды, перевязанные пожарным шлангом, кучи старых банок из-под краски…

Часть 1 Латаем ядро Ubuntu/Debian

Как мы уже сказали, вы заметите, что стандартное ядро большинства дистрибутивов Linux, как правило, изрядно перелатано, и Ubuntu – не исключение из этого правила. Чтобы все работало надежно, нужно получить новейший пакет с исходными кодами для ядра. Вы можете узнать, какое у вас стоит ядро, выполнив команду

```
uname -r
```

Она вернет строку с версией ядра. Для большей легкости получения результата работы вы должны использовать эту версию как основу вашей настройки. Если вы хотите использовать самое последнее ядро, то вы должны установить его, а затем получить соответствующие исходные тексты и наложить заплатки-патчи – имеет смысл сперва получить работающую систему, а потом уж ее исправлять.

Добудьте текущие исходники из вашего репозитория, например, так:

```
sudo apt-get install linux-source-2.6.20 kernel-package
libncurses5-dev fakeroot
```

Этот код извлекает исходные коды и некоторые другие вещи, необходимые для сборки ядра. Понятно, что во всех этих примерах вы должны заменить имя/номер пакета фактическим вариантом имени/номера ядра, которые вы хотите использовать в качестве кандидата для реформ в процессе данного урока.

Теперь, чтобы сэкономить на наборе команд, нам надо стать суперпользователем. Легче будет воспользоваться таким коварным приемом:

```
sudo bash
```

Это позволит нам избежать необходимости ставить sudo в начале каждой строки. Исходники ядра, как обычно, хранятся в **/usr/src/**, но Ubuntu немного ленив и не утрудился извлечь их для нас. Сделаем это сами:

```
cd /usr/src
tar xvjf linux-source-2.6.20
```

В этом месте принято использовать символьную ссылку linux на получившуюся директорию. Если у вас уже были установлены пакеты исходных текстов ядра, придется удалить эту ссылку и создать другую; в противном случае, выкиньте первую строку:

```
rm -f linux
ln -s linux-source-2.6.20-17 linux
cd linux
```

Предупреждение!

Бывает, что основной код дистрибутива рассчитан на некоторые расширения, предполагаемые доступными в ядре – и простая пересборка типового ядра может испортить часть системного ПО на настольных дистрибутивах вроде Ubuntu. Лучший (или, по крайней мере, простейший) способ проверить, что все будет работать – проведение собственных экспериментов над стандартным ядром для вашего дистрибутива. Не исключено, что некоторые облюбованные вами заплатки не заработают, поскольку противоречат существующим, но такова цена стабильности.

своe сам!

Кроме того, было бы полезно использовать текущий файл конфигурации для этого ядра. Таким образом мы можем убедиться, что мы по крайней мере, начинаем с конфигурации, которая работает! Ubuntu хранит конфигурацию в директории **/boot/**, поэтому просто скопируйте соответствующий файл:

```
cp /boot/config-2.6.20-17 ./.config
```

Теоретически, теперь можно собрать ядро, но оно, естественно, получится таким же, какое было – проку от этого немного. На самом деле мы хотим наложить кое-какие заплатки.

Применение заплаток

Ядра Ubuntu и Debian не слишком различаются. Большинство кода, специфического для Ubuntu, гуманно отделено в подкаталог ubuntu; это несколько упрощает нашу задачу. А то, что мы сейчас делаем, латая непосредственно исходники ядра, применимо почти к любому дистрибутиву или типовому ядру с www.kernel.org, так что этот этап достаточно общий.

Прежде чем браться за дело, вы должны уяснить, что латание исходников ядра – штука достаточно проблематичная. Бывает, что заплатки не приходятся ко двору, а иногда – что, возможно, даже хуже – оказываются успешны лишь частично. По возможности (т.е. при наличии места на диске), прежде чем пускаться в исправления, хорошо бы клонировать «хорошее» дерево исходного кода в другой каталог. Если все пойдет до ужаса наперекосяк, вы просто вернете чистую копию.

Первый шаг – загрузить файл заплатки. Убедитесь, что версия заплатки подходит к версии ядра, которое вы хотите латать. Обычно, если заплатка более ранней версии, вам может повезти в том, что код соответствующего раздела не менялся от версии к версии, но для масштабных заплаток – например, повышающих эффективность чего-то – как правило, нужна конкретная версия ядра.

Скачайте заплатку, скопируйте ее прямо в директорию **/usr/src/linux** и распакуйте (обычно это текстовый файл, сжатый *gzip* или *bzip*). По причинам, аналогичным приведенным ранее, можете сохранить загруженные заплатки где-нибудь еще, на случай если вам придется обновить эту директорию. Теперь протестируйте заплатку:

```
patch -p1 - - dry-run < patchfile-2.6.20
```

Вы получите вывод на консоль прогресса наложения заплатки и список всех сбоев. Он должен выглядеть примерно так:

```
patching file drivers/block/Kconfig
Hunk #1 succeeded at 464 (offset 8 lines).
patching file drivers/block/Makefile
patching file drivers/block/fcache.c
patching file fs/ext3/super.c
Hunk #3 succeeded at 709 (offset 40 lines).
Hunk #4 succeeded at 758 (offset 40 lines).
Hunk #5 succeeded at 786 (offset 40 lines).
Hunk #6 succeeded at 1089 (offset 40 lines).
Hunk #7 succeeded at 1446 (offset 40 lines).
Hunk #8 succeeded at 1454 (offset 40 lines).
Hunk #9 succeeded at 1533 (offset 44 lines).
```

Вам нужно проследить за строками, которые выглядят примерно так:

```
patching file mm/filemap.c
Hunk #1 succeeded at 454 (offset 20 lines).
Hunk #2 succeeded at 857 with fuzz 1 (offset 64 lines).
Hunk #3 FAILED at 1078.
```

Почему произошел сбой? Обычно из-за несовместимости заплаток, особенно если вы используете ядро, которое латали до вас. Иногда можно вычислить, почему заплатка слетает (вы можете увидеть сообщение о сбое, уже неплохо для начала). Ошибочные заплатки генерируют файл ***.rej**, там перечислен список кусков ["hunks"] кода, где имел место сбой.

У нас нет способа объяснить вам, что идет неправильно, но анализ файла **.rej** даст здесь пищу для ума. Часто бывает, особенно в случае заголовочных файлов, что из-за добавления новых определений (возможно, другой заплаткой), сместилась предполагаемая начальная точка вашей заплатки. Такие вещи легко исправляются ручным редактированием. Убедившись, что заплатка работает, примените ее:

```
patch -p1 - - dry-run < patchfile-2.6.20
```

Если наложение прошло нормально, следующим шагом должно быть конфигурирование ядра. Запустите *make xconfig* или другое приложение, которое вы используете, чтобы настроить исходный код ядра. Конфигурация теперь может включать новые опции, в зависимости от установленных заплаток. Так как вы скопировали *config* для немодифицированного ядра, вам придется найти и вручную установить требуемые опции, в зависимости от примененной заплатки. Подобные опции имеют не все заплатки, но многие.

Когда конфигурации завершена и записана, следующий шаг – собрать пакет. Для этого просто выполните следующие команды:

```
make-kpkg clean
fakeroot make-kpkg --initrd --append-to-version=-cust kernel_
image kernel_headers
```

Часть --append-to-version позволит отличить ваше самодельное творение от реальных версий пакетов из релиза – что очень упрощает жизнь, поверьте.

Этот шаг отнимает очень много времени (если только у вас не суперЭВМ), а в итоге вы получите два пакета: в одном – ядро, в другом – заголовочные файлы. Установите оба, командами

```
dpkg -i linux-image-2.6.20.xx-cust.xxxxx.deb
dpkg -i linux-headers-2.6.20.xx-cust.xxxxx.deb
```

Они заодно позаботятся о записи новых строк в загрузчик *grub*, сделав ваше ядро рабочим. Осталось только перезагрузиться, не позабыв выбрать новое ядро в загрузочном меню – а теперь надейтесь, что оно и вправду заработает! »

RPM



Некогда разработанный Red Hat для Red Hat Linux, RPM Package Manager вначале назывался Red Hat Package Manager. Название RPM означает две вещи: свободный программный инструмент, который устанавливает, обновляет, удаляет, проверяет и опрашивает программное обеспечение, и связанный с ним формат файла **.rpm**. Чтобы избежать путаницы в ходе этого урока, условимся, что заглавная аббревиатура означает менеджер пакетов, а строчная – файлы или их расширения. RPM поддерживается Fedora, Red Hat Enterprise Linux (и производных от нее CentOS и ASPLinux), Linux SUSE / OpenSUSE, Mandriva Linux (и его производной PCLinuxOS), PS2 Linux, Red Flag Linux, ArkLinux, Yellow Dog Linux, PLD Linux Distribution и Scientific Linux. RPM также портирован на некоторые другие ОС, например, IBM AIX (с версии 5) и Novell NetWare (с версии 6.5 SP3).

Часть 2 Латаем ядро Fedora



Настройка ядра

Многие люди страдают иллюзией, что удаление неиспользуемых функций ядра делает его быстрее и эффективнее. В общем случае это не так. Чуть быстрее будет сборка, чуть компактнее получится ядро, но существенной разницы вы не добьетесь, разве что на очень специфическом оборудовании.

Многие также имеют проблемы при отборе драйверов. Несомненно, если убрать лишние драйверы, не используемые в вашей системе, выигрыш будет немалым, но при обновлении своего компьютера или при переносе диска на другую машину, например, вы можете обнаружить, что сэкономленные память и время того не стоит.

Единственное место, где это будет хорошей идеей – ноутбук, тут вы в достаточной степени можете быть уверены в том, что основное оборудование вряд ли изменится.

В старые времена, Red Hat Linux и даже ранние версии Fedora компилировали ядро «обычным» способом: вы загружали пакет *kernel-source*, который сваливал все необходимое для сборки Fedora-совместимого ядра в директорию **/usr/src/kernel-2.xx.xx/**.

Однако, это, в некотором роде, не имело смысла, так как для правильной сборки ядра на RPM-системе важно, чтобы версия ядра была известна базе данных RPM (в целях соответствия модулей и т.д.). И если вы хотите собрать исправленное ядро Fedora, единственно верный способ сделать это – добраться до rpm-пакета с его исходниками (например, **kernel-2.x.x.i386.src.rpm**).

Отсюда следует, что вам нужны инструменты для сборки rpm… не пугайтесь, это не так сложно. Во-первых, переключитесь на суперпользователя и введите следующую команду

```
yum install fedora-rpmdevtools
```

Она просто установит последнюю версию инструментов построения rpm для Fedora – скриптов, макросов и команд, используемых при генерации **rpm** из файла **spec** и исходного кода – а также очень полезный для нас скрипт, который создаст директории, необходимые для сборки ваших доморощенных файлов, потому что использовать для этого «реальные» директории **rpm** (в **/usr/src/redhat**) – не очень хорошая идея. Затем, переключившись на обычную учетную запись пользователя, введите команду

```
fedora - buildrpmtree
```

что создаст копию каталога **/usr/src/redhat/** для сборки rpm-файлов под именем **rpmbuild**, поместив ее в домашний каталог пользователя.

Следующий этап – извлечение rpm-файла ядра, используемого Fedora. Вы должны найти его в дереве исходников вашего зеркала Fedora или репозитория (увы, из *yum* его установить нельзя). Скачайте этот файл в учетную запись пользователя. Как правило, в Fedora пользователям не дозволяется устанавливать **rpm**, но так как у вас установлены инструменты сборки и имеется корректное дерево, то при установке rpm-файла все, что полагается, попадет в директорию **rpmbuild**.

```
rpm -ivh kernel-2.6.21-3913.fc7.src.rpm
```

(подставьте свой номер версии). Скорее всего, вы увидите сообщения об ошибках, связанные с несуществующими пользователями, но их можно спокойно проигнорировать.

Теперь ваша директория **rpm** содержит все необходимые файлы. Вы обнаружите, что файл **spec** для ядра создан в **rpmbuild/SPECS**, а исходники ядра – в **SOURCES**, вместе со всеми файлами примененных к нему заплаток (поскольку практически все дистрибутивы не пользуются типовым ядром).

Выбираем заплатку

На первом этапе латания ядра, как ни странно, нужно скачать требуемую заплатку. Вы должны убедиться, что она применима к конкретной версии ядра. Код ядра меняется, и может случиться, что заплатка уже неактуальна. Конечно, утилита для наложения заплаток предупредит вас о любых проблемах, потому что интересующая вас часть ядра могла и не меняться между версиями, но лучше лишний раз убедиться, что номера версий согласуются.

Переименуйте ваш файл заплатки должным образом и поместите его в каталог **SOURCES** к другим заплаткам.

Чтобы применить заплатку для ядра, нужно включить ее в файл спецификации пакета. Итак, сдуйте пыль с вашего любимого текстового редактора и откройте файл **kernel.spec**. Найдите раздел с заплатками, который начинается с Patch1: xxxxx. Существует определенный порядок наложения заплаток, и вам нужно прочитать соответствующие примечания и убедиться, что вы добавили запись в нужную часть файла.

После добавления строки ищите дальше место в файле, где заплатка фактически применяется. Вы увидите строки типа:

```
Applypatch kernel2.6-23.make-things-work.patch
```

Вводите собственные строки в правильной последовательности: например, если ваша заплатка 4005, ее нужно ввести после строки patch4004.

Запасной вариант



Собрать ядро с нуля в наше время непросто, если вы хотите красивой сыгранности с установленным дистрибутивом. Но иногда это единственный способ получения последней версии драйвера или функции, которая нужна для работы ПРЯМО СЕЙЧАС! Если вы имеете привычку пересобирать и латать ядро, вы можете обнаружить, что процесс пошел неправильно, и в результате вам придется спасать свою загрузку от ядра-хулигана. Как и с данными любого другого рода, важно предусмотреть запасной вариант: стоит всегда иметь работающую запасную запись в загрузчике, или, по возможности, экспериментировать с ядрами на виртуальной машине.

Вы должны также вернуться к началу файла и изменить строку, начинающуюся с

```
#% define buildid .local
```

Удалите # и поставьте что-то вместо .local (но начинающееся с .). Это будет расширение, используемое в имени нового ядра и помогающее отличить его от стандартного. В этом пункте стоит протестировать сборку, чтобы убедиться, что все работает: команда

```
rpmbuild -bp kernel-2.6.spec
```

проведет вас через этот процесс. Любые ошибки будут выведены в консоли. Просмотрите их на предмет обнаружения конфликтов с вашей заплаткой – если таковые имеются, то либо забудьте о ней, либо найдите обновленную версию, способную работать с этим ядром.

Следующий шаг – сконфигурировать ядро. Вы сможете найти его исходные коды в директории **~/rpmbuild/BUILD/kernel-xxx/kernel-xxx/**. Перейдите туда и скомандуйте

```
make xconfig
```

для запуска программы конфигурации. Как и в случае с Ubuntu, тут необходимо включить новые опции, предоставляемые заплаткой. Чтобы завершить процесс, выполните, оставаясь в этой же директории,

```
make rpm
```

Эта команда создаст rpm-пакеты как с исходными кодами, так и с двоичными файлами, которые вы найдете в директории **rpmbuild/RPMs/i386/**, если не укажете другую архитектуру. Можете использовать обычные инструменты RPM для их установки! LXF

Сборка на Mandriva

В случае Mandriva, вы можете либо идти путем RPM (как в Fedora), или более стандартным маршрутом ручного наложения заплаток на исходные коды. Mandriva использует систему RPM, как и Fedora. Если вы загрузите файл **src.rpm** для ядра и инструменты Mandriva для создания пакетов, то вы можете построить rpm таким же образом.

Кроме того, можно загрузить пакет с исходниками ядра для вашей версии Mandriva и вручную наложить на них заплатки, как в случае Ubuntu или типового ядра, а затем замкнуть круг командой *make install*, обновив все. Недостаток второго подхода – система RPM больше не будет знать, какое ядро у вас установлено.

Где взять заплатки?

Ну, вообще-то изобилие заплаток имеется прямо на www.kernel.org. Они предназначены для применения в типовом ядре, но вполне могут сработаться и с версией ядра вашего дистрибутива. Особо интересны заплатки mm: это тестовые версии новых функций, предназначенные для включения в основную ветку ядра.

Главное меню и

ЧАСТЬ 3 Создать форму и набросать на ней пару кнопок и поле ввода – еще не значит написать настоящее бизнес-приложение! Пользователи ждут меню, и **Андрей Паскаль** знает, где его взять.



Наш эксперт

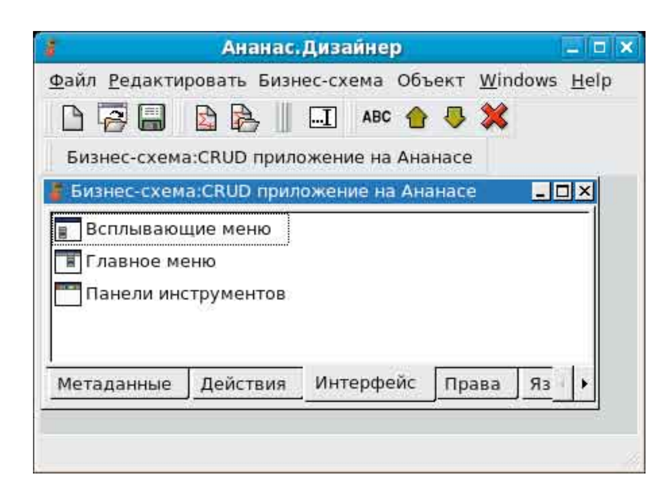
Андрей Паскаль использует Linux более семи лет и играет роль координатора проекта **Ананас**. Он также знает ответ на популярный у всех новичков вопрос: «Какой дистрибутив Linux мне выбрать?».

Если вы пропустили второй урок (LXF98), обязательно поинтересуйтесь его содержанием, чтобы материал настоящего выпуска был проще для восприятия. Мы будем использовать результат, полученный в прошлый раз – CRUD-приложение, позволяющее управлять справочником контрагентов.

Итак, продолжим. Мы уже в состоянии написать простое приложение на платформе *Ананас*, но владеем еще далеко не всеми приемами создания пользовательского интерфейса. Как известно, к нему относятся не только окна диалогов, но и различные виды меню. Традиционно, пользователь ожидает увидеть в любой программе такой элемент интерфейса, как главное меню. Разумеется, пользователи *Ананаса* не являются исключением. Поэтому сегодня мы научимся использовать средства среды разработки платформы *Ананас* для создания красивого и современного главного меню нашей бизнес-схемы.

Собираем каркас

За проектирование главного меню отвечает Редактор меню в Дизайнере Ананаса. Открывается он выбором вкладки Интерфейс окна редактирования свойств бизнес-схемы (**Рис. 1**). Редактор умеет работать с тремя видами меню: главное меню, меню в виде панели инструментов, дополняющее главное меню кнопками быстрого запуска наиболее востребованных функций, и всплывающие контекстные меню.



› Рис. 1. Редактор меню «живет» по адресу, обозначенному вкладкой «Интерфейс».

Lingua franca

Благодаря появлению средств разработки приложений, автоматизирующих учетную деятельность, программисты и пользователи наконец-то сумели найти общий язык: применяя термины «Документ», «Журнал», «Справочник», «Регистр», они отныне подразумевают одинаковые объекты! Конечно, у обеих сторон всегда останется свой жаргон «для внутреннего употребления». Например, в нашем цикле статей о платформе Ананас попадаются слова вроде «Действие», «Шаблон», «Модуль документа», «Модуль формы» и т.п., потому что статьи адресованы, в основном, разработчикам. Не горюйте, если смысл этих слов для вас темен: бизнес-схемами Ананаса можно успешно пользоваться, и не забираясь в программную кухню.

Для наших учебных целей мы создадим меню, напоминающее меню бизнес-схемы «Оперативный учет», поставляемой в дистрибутиве Ананаса. Разница будет заключаться в уменьшенном количестве пунктов и в том, что только два из них – Выход и Справочник контрагентов – будут выполнять полезную работу. Остальные пункты меню мы запрограммируем на вывод в окно сообщений текста, свидетельствующего о назначении выбранного пункта.

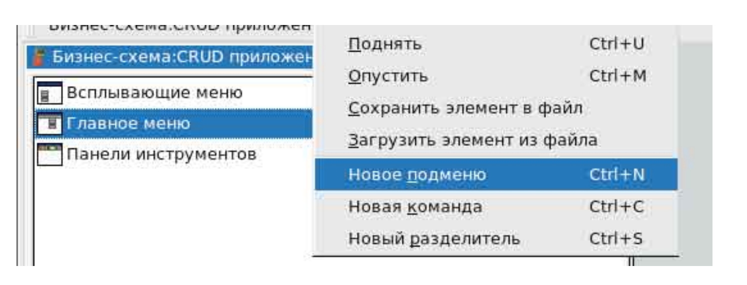
Чтобы не было путаницы с терминологией, хочу сразу пояснить, что главное меню состоит из элементов, которые могут называться «подменю», «команда» или «пункт меню». Можно считать эти термины синонимами, так как пункт меню пользователь рассматривает как соответствующую команду для выполнения. Команда же может не делать ничего, кроме открытия новых пунктов, которые принято называть подменю. Поэтому такая команда/пункт меню может тоже называться подменю.

Последовательность действий, приводящих созданию меню, выглядит следующим образом (**Рис. 2**). В контекстном меню, вызванном нажатием правой кнопки мыши в момент, когда ее указатель завис над надписью Главное меню, выбираем пункт Новое подменю. Главное меню пополнится пунктом Новое подменю. Однако нам нужен пункт

Месяц назад Мы создали простое CRUD-приложение средствами Ананаса.



панель инструментов



› Рис. 2. Создание пункта главного меню.

Программа – поэтому, выбрав мышью Новое подменю и нажав функциональную клавишу F2 на клавиатуре, меняем надпись на Программа, а затем нажимаем Enter.

Точно так же добавляем пункт меню Справочники. Пожалуй, хватит. Помимо этих двух пунктов, меню любой бизнес-схемы будет автоматически дополнено сервисными пунктами Помощь и Окна. Таким образом, главное меню будет состоять из пунктов Программа, Справочники, Помощь, Окна.

Заселить и оживить!

Разумеется, этого нам совершенно не достаточно. Пользователь привык видеть в главном меню не просто набор пунктов, а подменю с подпунктами. Не будем обманывать его ожидания – добавим подменю в созданные нами пункты главного меню.

Пусть подменю Программа содержит ровно один пункт – Выход. Зато это будет полезный пункт подменю, вызывающий завершение работы *Ананаса*. А вот подменю Справочники мы наполним четырьмя пунктами и одним разделителем. В нашем небольшом учебном меню разделитель можно было бы и не ставить. Но без разделителей не обойтись в любом реальном более-менее сложном приложении, где требуется группировка большого количества пунктов меню по смысловой нагрузке для облегчения работы пользователя. Поэтому не будем забывать про разделители!

Из контекстного меню над пунктом Программа следует выбрать «Новая команда». В результате откроется окно редактирования свойств пункта меню, где следует занести слово Выход в поле Текст меню и любое удобное вам название команды в поле Имя. В редакторе меню вы будете работать с идентификатором из поля Имя, пользователь же будет видеть надпись на пункте меню из поля Текст меню. Если вы сомневаетесь, какое значение следует задать полю Имя – мой совет: такое же, какое записано в поле Текст меню, то есть Выход.

На этом можно завершить работу по настройке пункта Выход подменю Программа и закрыть редактор его свойств. Позже мы вернемся в этот диалог, чтобы завершить настройку.

Далее следует таким же образом добавить и настроить пункты подменю Справочники.

Сначала добавьте пункты Справочник товаров и Справочник контрагентов. Затем вставьте разделитель, выбрав Новый разделитель из того же контекстного меню. Последними добавьте пункты меню Справочник складов и Справочник констант. Для выполнения этих действий полезно будет знать, что всякий раз, когда вы хотите добавить новый пункт подменю, контекстное меню необходимо вызывать, выделив имя подменю, а не имя одного из его пунктов. То есть, добавляя пункты в подменю Справочники, всякий раз необходимо вызывать контекстное меню, наводя указатель мыши на строку со словом Справочники.

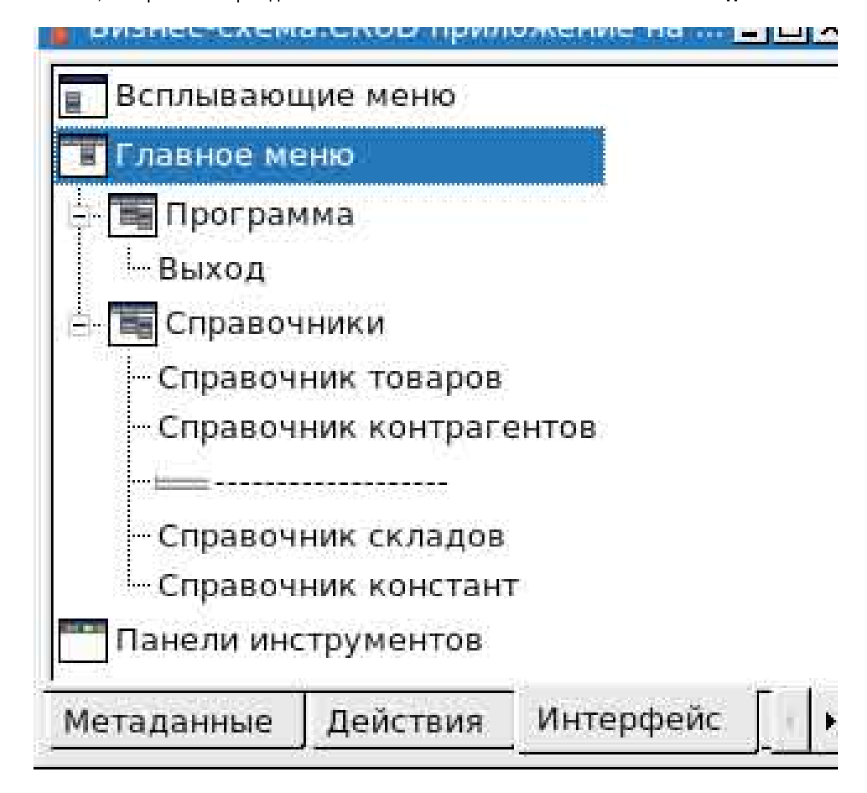
Если потребовалось поменять порядок следования пунктов подменю, это всегда можно сделать, воспользовавшись действиями Поднять и Опустить из контекстного меню. Результат вашей работы должен выглядеть примерно так, как показано на **Рис. 3**.

Половина дела сделана. Если сейчас сохранить выполненную работу, то есть записать бизнес-схему (немедленно сделайте это, если вы еще ни разу не нажимали на иконку-дискету в главном меню Дизайнера) и запустить ее на исполнение в *Ананасе*, то все подменю и их пункты будут видны и доступны. Однако их выбор не будет вызывать никаких действий. И это совершенно логично: ведь мы еще не указали *Ананасу*, какие действия должны выполняться при выборе того или иного пункта главного меню.

Делается это довольно просто, но сначала давайте разберемся, что такое «Действие» в терминологии среды разработки *Ананаса*. Жизненное пространство этого понятия ограничено средой разработки, то есть с ним сталкивается только программист, создающий бизнес-приложение. Пользователь бизнес-приложения, выполняемого *Ананасом*, совершенно огражден от использования этого понятия.

»

› Рис. 3. Состав главного меню задан.



Пытаясь придумать формальное определение, можно получить что-то типа «Действие – это именованный объект, содержащий указания о выполняемых операциях, либо исполняемый код на языке программирования платформы *Ананас*, предназначенный для использования при настройке команд меню и других объектов бизнес-схемы».

Создадим пару действий, чтобы оживить наше меню и заставить его работать. Благо, сделать это совершенно несложно: нам поможет редактор действий, расположенный рядом с редактором меню.

По многочисленным просьбам тех, кто руководствуется правилом «Прежде чем нажать кнопку Пуск, найди кнопку Стоп», первым действием, которое мы запрограммируем, будет выход из программы.

Выбрав вкладку Действия, наведем мышь на строку с надписью Действие и выберем в контекстном меню, вызванном правой клавишей, , пункт Новое действие. В открывшемся окне присвоим полю Имя значение Закрыть Ананас, а полю Тип – значение скрипт, и, выбрав вкладку Выполнить, впишем волшебную команду, вызывающую завершения исполнения *Ананаса*:

```
sys.Exit();
```

Для отладочных целей вместо реальных команд рекомендую пользоваться функцией вывода сообщения в окно сообщений.

```
sys.Message(0,"Вызван пункт меню");
```

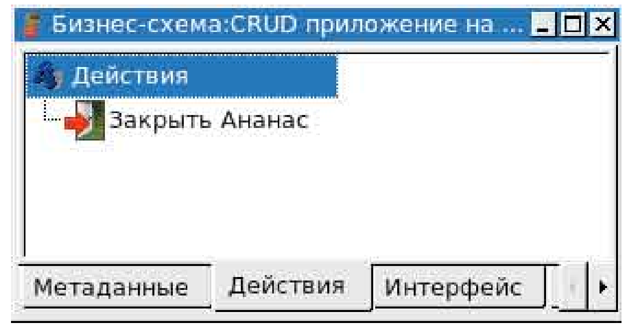
Попробуйте, например, сами определить отдельное действие для каждого из пунктов Справочник складов и Справочник констант подменю Справочники, чтобы при выборе пункта меню в окно сообщения выводилось сообщение о том, какой пункт меню был выбран.

С настройкой действия покончено, результат достигнут. Но! Есть возможность украсить пункт Выход подменю Программа каким-нибудь графическим изображением (значком). Разумеется, если у вас есть подходящий файл с картинкой. Задать значок можно на общей вкладке диалога редактирования свойств действия. Я использовал такой . После закрытия окна редактор действий принял вид, показанный на **Рис. 4**.

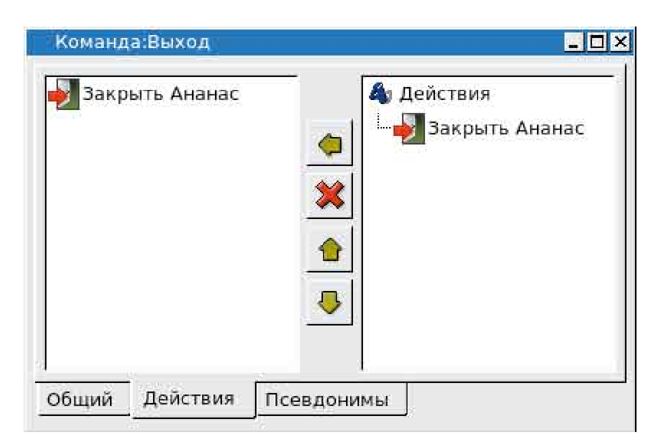
Теперь осталось связать только что созданное действие с пунктом Выход подменю Программа. Для этого возвращаемся в редактор меню, открываем диалог редактирования команды Выход двойным щелчком по соответствующей строке и на вкладке действия переносим действие Закрыть Ананас из правого списка в левую часть. Результат показан на **Рис. 5**. Окно можно закрыть и сохранить бизнес-схему, чтобы попробовать меню в деле. Теперь выбор пункта Выход главного меню будет закрывать программу.

Идем дальше. Не менее полезным мы можем сделать пункт Справочник контрагентов подменю Справочники, настроив его на открытие справочника контрагентов. Тогда Справочник контрагентов, открываемый по умолчанию при запуске нашего CRUD-приложения, можно будет смело закрывать, так как для его повторного открытия больше не потребуется перезапускать *Ананас*. Достаточно будет выбрать соответствующий пункт меню, чтобы справочник открылся точно так же, как он открывается при запуске *Ананаса*.

Итак, добавим новое действие, скажем, Открыть справочник контрагентов. Его мы определим новым способом, не так, как действие Закрыть Ананас. Нам не потребуется прописывать ни одной команды. Точнее сказать, мы могли бы придумать скрипт, осуществляющий открытие справочника контрагентов, но в данном случае есть более простой путь. И мы им воспользуется.



› Рис. 4. Действие "Закрыть Ананас" готово действовать!



› Рис. 5. Редактор свойств пункта меню. Привязка действия к команде.

Выбрав вкладку действия, создадим новое, введя в поле Имя текст Открыть справочник контрагентов и оставив в поле Тип значение, стоящее там по умолчанию – стандартная команда. И после выбора вкладки Выполнить, сделаем следующее. Полю Действие зададим значение редактировать. В поле Объект укажем, какой объект нужно редактировать – Справочник контрагентов. В поле Форма зададим, какую следует использовать форму для редактирования Справочника контрагентов – Встроенный редактор. Остальное за нас сделает *Ананас*. Так что можно закрыть диалог и, сохранив бизнес-схему, запустить *Ананас* для проверки нашей работы. Если все сделано правильно, мы можем перейти к последней части нашего урока – работе с Панелью инструментов.

Средства быстрого доступа

Панель инструментов предоставляет пользователю набор кнопок быстрого доступа к наиболее часто используемым функциям приложения. Эти кнопки обычно расположены непосредственно под главным меню, имеют квадратную форму и снабжены изображением (значком), символизирующим выполняемое действие.

Ананас позволяет создавать множество таких панелей, которые пользователь по своему усмотрению может перемещать, располагать по периметру главного окна приложения или выносить за его пределы в виде самостоятельного окна панели инструментов.

Создание панели инструментов происходит в редакторе меню Дизайнера Ананаса, использование которого мы уже освоили при создании главного меню. Однако, есть небольшое отличие: если команды можно было добавлять непосредственно в главное меню (минуя этап создания подменю), то сейчас нам необходима хотя бы одна панель инструментов. Получите ее, активировав пункт «Новое подменю» из контекстного меню. Теперь можно добавлять команды, которые будут представлены пользователю в виде кнопок.

Разумеется, решение о том, какие функции приложения выносить на панели инструментов, принимает разработчик. Для нашего учебного примера соорудим простую панель из двух кнопок, которые, как нетрудно догадаться, будут выполнять выход из приложения и открытие нашего справочника контрагентов.

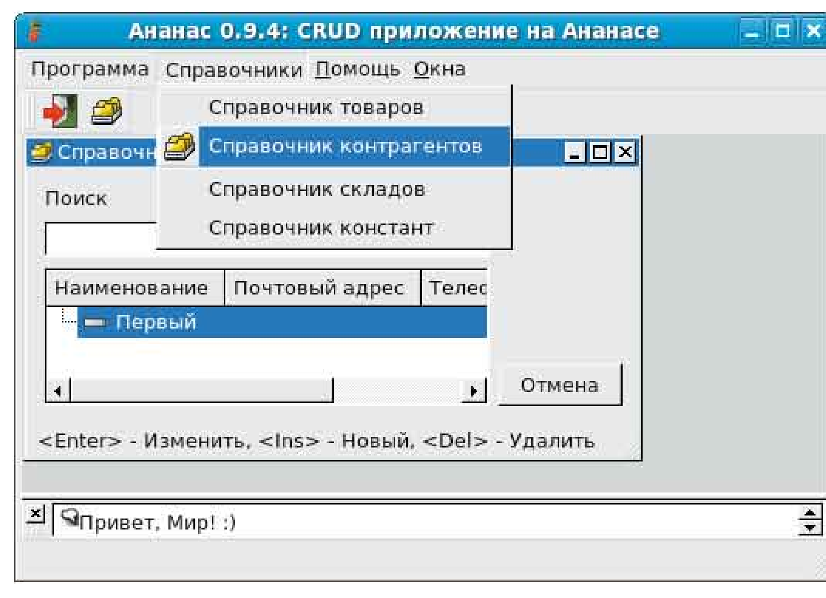
Приятным моментом здесь является то, что определять новые действия нам не придется – мы просто воспользуемся действиями, определенными ранее для главного меню.

Определив панели инструментов, мы завершили настройку пользовательского интерфейса нашего приложения. **Рис. 6** демонстрирует, как выглядит наша работа в Дизайнере. **Рис. 7** показывает, как будет видеть нашу работу пользователь бизнес-схемы. LXF

Через месяц Мы займемся проектированием документов и подготовкой печатных форм



› Рис. 6. Результат – главное меню и панель инструментов определены..



› Рис. 7. Результат – CRUD-приложение с красивым главным меню и двухкнопочной панелью инструментов.





Экстракт кофе

ЧАСТЬ 11 Наша серия, увы, подходит к концу, но на десерт **Александр Бабаев** припас нечто особенное – ароматные зерна Enterprise Java Beans



Все предыдущие статьи содержали огромное количество полезной информации. Мы рассмотрели алгоритмы, структуры, принципы работы, множество библиотек, так или иначе связанных с созданием серьезных приложений на Java. И пусть вас не смущает то, что примеры были простые. В основе больших, серьезных, важных и сложных приложений лежит именно то, о чем мы говорили.

Теперь вы уже готовы узнать, что обозначает загадочная аббревиатура EJB3. Расшифровывается она как «Enterprise Java Beans, version 3» и содержит внутри огромный мир, окошко в который мы сегодня приоткроем.

Почему именно EJB3?

Дело в том, что это стандарт. Тройка в названии указывает, что были еще версии один и два, также была версия два-точка-один… Но только текущая, третья версия действительно является великолепным инструментов для борьбы с хаосом корпоративных систем. Все предыдущие версии строились по такому принципу: «Мы (**Sun/IBM/** и пр.) тут собрались, посовещались и решили, что вы (разработчики) будете использовать вот это… (EJB2.1)». Было круто, но разработчики хоть и использовали, но плевались (странно, правда?). EJB3 создавалась иначе. Те же люди собрались, посмотрели на то, как работают программисты, какие есть библиотеки, удачные решения, технологии… И, выбрав лучшее и добавив свой (огромный) опыт, выдали третью версию спецификации.

Получившийся стандарт хорош. Хорош и простотой (можно обойтись без специфических XML, которых в предыдущих вариантах были сотни), и привычностью (*Hibernate* использовали? Нет? Ну, это теперь reference implementation, то есть стандартная реализация для *JPA1*, части EJB3), и заменой старых неудобных частей на новые, «блестящие и шелковистые».

Для чего оно?

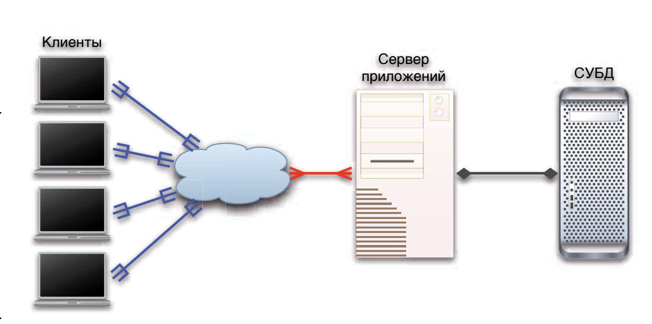
Большие системы никогда не создаются в одиночку. Существуют люди, которые зовутся архитекторами: они придумывают систему. Система обычно состоит из блоков. Блоки, в свою очередь, состоят из других блоков… Блоки отвечают за совершенно разные вещи: за хранение бизнес-объектов, за просчет алгоритмов, за управление элементами системы, и так далее. Разрабатываются эти составные части системы разными людьми, часто совершенно не связанными друг с другом.

В таких условиях нужен стандарт, который обеспечивал бы, чтобы блок, написанный в Индии, и блок, написанный в России, заработали вместе. Можно этот стандарт каждый раз придумывать заново, но на это никогда нет времени. Лучше использовать EJB3.



Общая структура EJB-проекта

Упрощенная схема проекта приведена на **Рис. 1**.



› **Рис. 1. Трехзвенная структура EJB-проекта.**

Это так называемая трехзвенная структура. Она проста, и для более сложных систем может разрастись до четырех-, пяти-, *n*-звенной. Серверов приложений может быть кластер, СУБД тоже может представлять кластер с распределением нагрузки и резервированием, и так далее.

В качестве СУБД сгодится практически любая: у нас это будет *MySQL*, но, грамотно используя JPA (об этом чуть дальше), СУБД можно сменить хотя и за ненулевое, но вполне приемлемое время. Клиенты могут быть либо тонкими (браузер), либо полноценными приложениями (и даже не обязательно на Java).

А вот про сервер приложений поговорим подробнее. Это именно то место, где соединяются те самые блоки, написанные в Индии, России, Китае, США и так далее. Чтобы все они работали вместе, написаны специальные приложения, которые обеспечивают связь с СУБД, предоставляют огромное количество стандартных API для работы блоков, дают возможность эти блоки выгружать и загружать без перезапуска сервера, контролировать их исполнение… и много чего еще. Сервера приложений есть как коммерческие (*IBM WebSphere*, *BEA WebLogic*), так и бесплатные (*GlassFish*, *JBoss*, *IBM WebSphere Community Edition*). Мы посмотрим поближе на *JBoss*, который, помимо прочего, продвигается Red Hat и распространяется в составе Red Hat Enterprise Linux.

JBoss

Итак, сервер приложений. Он состоит из огромного количества блоков, интегрированных вместе. Некоторые нам уже так или иначе знакомы: например, *Tomcat* или подсистема RMI-подключений. Но есть и мно-

Месяц назад Мы узнали, что такое Struts и почему для Java существует так мало CMS-систем.

жество других: Hibernate, который отвечает за «связь с СУБД», система кэширования, кластеризации, распространения сообщений, транзакций (уровня приложения), …

Короче, система большая. Мы рассмотрим относительно небольшую часть, которая работает с EJB3, задержимся на Hibernate (и JPA) и немного поглядим на транзакции (JTA). Все рассмотреть, конечно, не успеем, поэтому после статьи приведен список литературы – выберите книжку по вкусу, чтобы заняться подробным изучением.

Так что же такое EJB?

Ну, во-первых, это технология, это уже понятно. Во-вторых, это Enterprise Java Bean (без «s») – то самое «зерно», то есть блок, из множества которых строится приложение. Блок стандартным образом упаковывается, и помещается в каталог *JBoss*, после чего (если все сделано правильно), *JBoss* подключает блок к системе.

То есть блок – это бин (bean). И наоборот. Бин в простейшем случае – это всего лишь класс, описанный и объявленный специальным образом. Например, пусть он считает площадь круга.

```
@Remote
public interface Calculator {
  public double getSquare(double aRadius);
}
```

Пока это только объявление, то есть интерфейс – то, что видит клиент. Причем от обычного интерфейса он отличается только словом @Remote (так называемой аннотацией). Оно обозначает, что клиенты, которые используют этот бин, могут находиться как на серверном компьютере, так и на любых других узлах сети. Если доступ извне локального компьютера не предполагается, то можно использовать аннотацию @Local, или вообще ничего не писать, так как интерфейсы считаются локальными по умолчанию. Умолчания – это одно из огромных достоинств EJB3, так как не нужно прописывать банальности, которые в крупных проектах превращаются в мегабайты ненужного кода.

Но где же считается сама площадь? В классе, который реализует интерфейс:

```
@Stateless
public class CalculatorBean implements Calculator {
  public double getSquare(double aRadius) {
    return Math.PI*aRadius*aRadius;
  }
}
```

Тут опять появилась аннотация, которая обозначает, что этот бин (да, это настоящий Enterprise Java Bean; да, больше – кроме упаковки в jar – не нужно вообще ничего) является Stateless-бином, то есть не сохраняет состояние в процессе работы. Клиент вызывает метод, метод выполняется, и следующий метод ничего не будет знать о предыдущем исполнении.

Теперь давайте посмотрим на клиент и, наконец, поставим *JBoss*, запустив наш бин.

Клиент будет немного «не в стиле EJB3». Правильный EJB3-клиент выглядит примерно так:

```
public class CalculatorClient {
  @EJB
  private Calculator _calculator;

  private void start() {
    System.out.println("Square for circle with radius 2.345 = " +
          _calculator.getSquare(2.345));
  }

  public static void main(String[] args) {
    new CalculatorClient().start();
  }
}
```

Но *JBoss* (который мы будем использовать для демонстрации) такого стиля (пока) не понимает, поэтому выйдет чуть-чуть подлиннее:

```
public class CalculatorClient {
  private Calculator _calculator;

  private void start() throws NamingException {
    _calculator = (Calculator) getInitialContext().
          lookup("CalculatorBean/remote");
    System.out.println("Square for circle with radius 2.345 = " +
          _calculator.getSquare(2.345));
  }

  private InitialContext getInitialContext() throws NamingException {
    Properties properties = new Properties();
    properties.put(Context.INITIAL_CONTEXT_FACTORY,
          "org.jnp.interfaces.NamingContextFactory");
    properties.put(Context.PROVIDER_URL, "localhost:1099");
    return new InitialContext(properties);
  }

  public static void main(String[] args) throws NamingException {
    new CalculatorClient().start();
  }
}
```

Разница, как можно заметить, в отсутствии аннотации @EJB. А заменяет ее тот код, который мы написали (получение контекста, а из него – ссылки на нужный нам бин).

Нужно сделать небольшое отступление. При поиске объекта мы использовали его имя: "CalculatorBean/remote". Почему именно так? И как вообще ищутся объекты? Для этого используется так называемая служба имен Java (JNDI) – нечто похожее на реестр RMI, который использовался для регистрации и поиска RMI-серверов. А имя присваивается автоматически и по умолчанию. В принципе, можно дать указания серверу приложений, как должен называться бин в системе.

Именно служба имен гарантирует, что бин, даже если его выгрузить и загрузить обновленный, будет найден остальными частями системы. А если при этом не изменять его интерфейс, то можно безболезненно улучшать реализацию, при этом не куроча всю систему и не навлекая гнев соратников по клавиатуре.

Осталось упаковать классы, скинуть их серверу приложений и посмотреть, что же из этого выйдет.

Первый бин

Для начала скачаем *JBoss*. Это можно сделать на страничке http://labs.jboss.com/jbossas/downloads/; берите стабильную версию (4.2.2.GA). Ее также можно найти на диске, прилагаемом к журналу.

Установка *JBoss*'а заключается в разархивировании его куда-нибудь. После этого он готов к работе (предполагается, что пути к **/bin/java** уже находятся в PATH).

Теперь сделаем бин. Для этого нужно скомпилировать наш интерфейс и класс (путь к *JBoss*'у, конечно, подставьте свой):

```
javac -cp ~/bin/jboss/server/default/lib/jboss-ejb3x.jar Calculator.java
javac -cp ~/bin/jboss/server/default/lib/jboss-ejb3x.jar CalculatorBean.
java
```

после чего запаковать все в jar-файл:

```
jar -c CalculatorBean.class Calculator.class > CalculatorEJB.jar
```

Теперь начинается волшебство. Запускаем JBoss:

```
jboss/bin/run.sh -c default &
```

(понятно, что путь к *JBoss*'овскому **run.sh** будет другой, понятно, что если требуется запустить его с возможностью закрыть терминал, нужно использовать nohup… это самый простой вариант) и кидаем jar-файл в каталог **/jboss/server/default/deploy**.

Все, через секунду бин «воткнут» в *JBoss*. Теперь можно запускать клиента. Компилируем его:

```
javac -cp ~/bin/jboss/server/default/lib/jboss-ejb3x.jar:. CalculatorClient.
java
```

Запускаем клиент, при этом подключая нужные библиотеки (все они есть в *JBoss*'е):

```
java -cp ~/bin/jboss/client/jbossall-client.jar:. client.CalculatorClient
```

»



» Получаем:

```
Square for circle with radius 2.345 = 17.275696541906616
```

Кто не верит, пусть пересчитает на калькуляторе.

Второй бин. JPA

Почувствовали, насколько это просто и быстро? Теперь попробуем что-нибудь сохранить в СУБД. Для этого немного настроимся. Подключать будем *MySQL*, которая запущена на локальной машине, с настройками по умолчанию (учетная запись root без пароля). Чтобы подключить СУБД, нужно положить специальный XML-файл в тот же каталог **jboss/server/default/deploy**. Файл будет называться **mysql-ds.xml**, а содержать будет следующее:

```
<?xml version="1.0" encoding="UTF-8"?>

<datasources>
  <local-tx-datasource>
          <jndi-name>jdbc/jpaPool</jndi-name>
          <connection-url>jdbc:mysql://localhost:3306/nisp</connection-url>
          <driver-class>com.mysql.jdbc.Driver</driver-class>
          <user-name>root</user-name>
          <password></password>
          <exception-sorter-class-name>org.jboss.resource.adapter.jdbc.vendor.MySQLExceptionSorter</exception-sorter-class-name>
  </local-tx-datasource>
</datasources>
```

Тут все просто: строка соединения нам попадалась уже при рассмотрении JDBC в LXF93 (можно ее усложнить, введя при необходимости всякие параметры), имя учетной записи и пароль – тоже вполне самоочевидные вещи. Еще указывается jndi-name, то есть имя, по которому этот пул соединений можно будет найти в недрах *JBoss*.

Положили? Теперь перезапустите сервер приложений:

```
jboss/bin/shutdown.sh -s jnp://127.0.0.1:1099 -S
jboss/bin/run.sh -c default &
```

Теперь мы готовы; начинаем писать EntityBean. Так называются классы-сущности, экземпляры которых хранятся в СУБД.

JPA, или Java Persistence API – это стандартный API, предназначенный для того, чтобы в удобном виде хранить Java-объекты в БД. При этом создается структура БД, создаются классы, в которых при помощи аннотаций прописываются связи между полями классов и полями таблиц БД, а остальное (преобразования, проверки, …) делает конкретная реализация JPA. В *JBoss* это Hibernate.

Итак, сделаем микробиблиотеку. Вот класс, который хранит в БД информацию о книге:

```
@Entity
public class Book implements Serializable {
  @Id
  @GeneratedValue(strategy = GenerationType.AUTO)
  public Long Id;

  public String Title;
  public String Author;
}
```

И все. Мы вновь использовали аннотацию, на сей раз @Entity. Все остальное – умолчания, а значит, поля класса Title и Author будут полями в таблице Book. Id – это идентификатор объекта, он уникален, и аннотацией мы указываем, чтобы JPA сам его создавал, когда будет нужно. Можно прописать и подробности: самостоятельно дать имена полям и таблице, указать типы полей и т.п., но пока оставим все это на усмотрение JPA.

Расширим микробиблиотеку, введя туда микрочитателей. Предположим, что один читатель может читать только одну книгу.

```
@Entity
public class Reader implements Serializable {
  @Id
  @GeneratedValue(strategy = GenerationType.AUTO)
  public Long Id;

  public String Name;

  @OneToOne
  public Book Book;
}
```

Аннотация @OneToOne обозначает, что связь между Reader и Book – один к одному. Физически эта связь будет реализована посредством внешнего ключа в таблице.

Возникает вопрос: а в какой таблице будет прописана связь (создан внешний ключ)? В Book? Или в Reader? Нужно как-то определиться, чтобы более четко понимать, что происходит в системе – иначе как ошибки-то искать? Для этого пропишем ссылку на Reader'а в книге, плюс укажем аннотацией, с какой стороны должна быть ссылка в БД. Теперь книга выглядит так:

```
@Entity
public class Book implements Serializable {
  @Id
  @GeneratedValue(strategy = GenerationType.AUTO)
  public Long Id;

  public String Title;
  public String Author;

  @OneToOne(mappedBy = "Book")
  public Reader Reader;
}
```

Параметр аннотации @OneToOne mappedBy показывает, что указанное поле в БД (в таблице Book) не существует, а извлекается из таблицы, на которую указывает связь, то есть Reader.

Третий бин

Создав бины для хранимых сущностей (Entity Beans), нужно сделать бин для работы с ними: извлечения из СУБД, сохранения, выдачи книги и возврата ее обратно в библиотеку.

Для экономии места не будем увлекаться всевозможными проверками (например, выдана ли книга повторно), сосредоточимся на JPA и EJB3.

Как и в предыдущий раз, сначала интерфейс:

```
@Remote
public interface Library {
  public void addBook(Book aBook);
  public void addReader(Reader aReader);
  public void giveBook(String aReaderName, Book aBook);
  public void returnBook(String aReaderName);
```

Как рекомендуется создавать бины

После деплоймента (установки и развертывания) бина микробиблиотеки, JPA автоматически создал все необходимые таблицы, связи между ними и так далее. Казалось бы, это счастье. Пишем код, JPA делает таблицы – все довольны… Но не так все просто.

Во-первых, зачастую таблицы уже есть. Во-вторых, далеко не всегда разработчики, которые создают бины, умеют хорошо создавать таблицы. В третьих, иногда таблицы и бины создают разные люди.

И это правильно. Нехорошо, если разработчик «и швец, и жнец, и на дуде игрец» (то есть хорошо, конечно, но где ж их взять в достаточном для пропитания количестве?). Правильный путь разработки структуры данных – когда сначала создается структура таблиц в СУБД, правильно прописываются связи, при необходимости реализуются триггеры и другие особенности, связанные с СУБД, и только потом на этой основе создаются бины, объекты, связи между ними. JPA рассчитан именно на такое применение и, в принципе, умеет не только создавать структуру СУБД по Java-файлам, но и наоборот, создавать классы по базе. Стоит это учитывать. Иначе часто получается, что «хотели как лучше, а получилось как всегда», или «что тут этот дурак JPA насоздавал – я совсем не то имел в виду; да и тормозит всё».

```
    public Reader getReader(String aReaderName);
    public Book getBook(String aBookTitle);
}
```

Потом его реализация (и на ней остановимся поподробнее):

```
@Stateless
public class LibraryBean implements Library {
    @PersistenceContext
    private EntityManager _entityManager;

    public Book addBook(Book aBook) {
        _entityManager.persist(aBook);
        return aBook;
    }

    public Reader addReader(Reader aReader) {
        _entityManager.persist(aReader);
        return aReader;
    }

    public void giveBook(String aReaderName, Book aBook) {
        getReader(aReaderName).Book = aBook;
    }

    public void returnBook(String aReaderName) {
        getReader(aReaderName).Book = null;
    }

    public Reader getReader(String aReaderName) {
        return (Reader) _entityManager.createQuery(
            "SELECT r FROM Reader AS r WHERE r.Name = :name").
            setParameter("name", aReaderName).getSingleResult();
    }

    public Book getBook(String aBookTitle) {
        return (Book) _entityManager.createQuery(
            "SELECT b FROM Book AS b WHERE b.Title = :title").
            setParameter("title", aBookTitle).getSingleResult();
    }
}
```

Тут есть два интересных момента. Первый – @PersistenceContext. Это так называемый контекст сохранения объектов, в котором обретается все то, что должно «жить» между перезапусками приложения (то есть записываемое в БД). Он нигде не инициализируется явно, так как эту работу берет на себя JPA.

У этого контекста есть несколько (не так много, как могло бы быть) методов, которые позволяют сохранять (persist) и удалять (remove) объекты. Но нет метода, который позволял бы обновлять объекты… Почему? Посмотрим более внимательно на метод giveBook, который «выдает книгу» читателю. Мы просто извлекаем читателя, присваиваем ему ссылку на книгу… и все. Контекст сохранения сам обновит объект в БД. Не правда ли, просто?

И последний интересный момент: запросы. Несмотря на то, что они очень похожи на *SQL*, это совсем не *SQL*. Это *JPQL* (Java Persistence Query Language), который оперирует не столбцами, но объектами. Например, предположим на минутку, что разные экземпляры одной и той же книги могут читать сразу несколько читателей (связь OneToMany). Тогда найти всех людей, которые держат книгу с определенным названием, можно так:

```
SELECT r FROM Reader AS r WHERE r.Book.Title=:title
```

На *SQL* нам пришлось бы выполнить JOIN, учесть внешние ключи и так далее. Здесь все это будет сделано автоматически.

Мало того, контекст заботится и о транзакциях. Он сам открывает транзакцию перед входом в методы и сам же закрывает ее после выхода. Естественно, используя JTA (Java Transaction API), это можно контролировать вручную, если есть необходимость.

Собираем, запускаем

Собрать jar-файл, чтобы положить его в *JBoss*, можно аналогичным образом: скомпилировать, собрать jar и поместить его в каталог deploy. Но при этом нужно сделать пару вещей, которые позволят JPA автоматически создать структуру таблиц при запуске бина. Для этого необходимо перейти в каталог **jboss/server/default/deploy/ejb3.deployer/META-INF** и в файле **persistence.properties** убрать комментарий (решетку) перед строкой

```
hibernate.hbm2ddl.auto=create.
```

После чего перезапустить *JBoss*, скинуть jar… бины встали на место. Попробуем простенький клиент для проверки. Основной код остался тем же, что и у клиента калькулятора, поменялся только метод start.

```
private void start() throws NamingException {
        _library = (Library) getInitialContext().
            lookup("LibraryBean/remote");

        Book book = new Book();
        book.Title = "Linux strikes back";
        book.Author = "Community org.";

        book = _library.addBook(book);

        Reader reader = new Reader();
        reader.Name = "Hacker I.A.";

        _library.addReader(reader);

        _library.giveBook("Hacker I.A.", book);

        reader = _library.getReader("Hacker I.A.");

        System.out.println("У читателя \"" + reader.Name + "\" " +
            "есть книга \"" + reader.Book.Title + "\"");
}
```

Запустите его и убедитесь, что все работает, как надо.

Вместо заключения

Заключения тут не получится. Мы рассмотрели только камень на вершине айсберга, который называется EJB3. Чтобы только перечислить, что же оно может, нужно раз в 5–10 больше места. А чтобы разобраться, нужно потратить не одну неделю. Поэтому хочется просто отметить, что, в отличие от стандарта EJB2, который использовать очень и очень трудно, EJB3 использовать можно и нужно. Есть надежда, что скоро все основные сервера приложений будут поддерживать EJB3 в полном объеме, что позволит быстро и просто разрабатывать сложные, надежные и производительные приложения, рассчитанные на работу в системах уровня предприятия. Если вы заинтересовались данной темой, и хотите узнать про EJB3 побольше, обратите внимание на врезку *Литература*. Удачного освоения! LXF

Литература

К сожалению, про EJB3 пока очень мало что написано на русском языке. Но на английском есть несколько книг, из которых хочется порекомендовать следующие:

» Raghu R. Kodali, Jonathan R. Wetherbee, Peter Zadrozny. *Beginning EJB 3 Application Development: From Novice to Professional* (ISBN 1590596714).

» Mike Keith, Merrick Schincariol. *Pro EJB 3: Java Persistence API* (ISBN 1590596455).

Книги написаны приличным языком, со знанием дела (авторы участвовали в разработке стандартов и работают в компаниях, которые сами используют EJB3), читать их достаточно легко.

Ну, и, конечно, нельзя не указать на сами стандарты:

» http://java.sun.com/products/ejb/
» http://java.sun.com/products/ejb/docs.html
» http://java.sun.com/javaee/technologies/persistence.jsp



D-Bus: шины для вашего Linux

Уже успели подумать о зимней резине или переломах и вывихах? Возвращайтесь обратно в виртуальный мир – **Андрей Боровский** имеет в виду шину для обмена данными между настольными приложениями!



Что такое D-Bus? Самый простой ответ – еще одна система межпроцессного взаимодействия (Interprocess Communication или IPC). Ключевые слова здесь «еще одна». Высокоуровневых систем IPC для Unix/Linux существует много. Помимо систем высокого уровня Unix обладает развитыми средствами IPC низкого уровня (сокеты, каналы), каковые успешно используются многими приложениями напрямую. Зачем тогда нам может понадобиться D-Bus? Эта система замышлялась группой *FreeDesktop.org* как средство IPC, не зависящее от типа рабочего стола, призванное заменить как DCOP в KDE, так и CORBA/Bonobo в GNOME. Вытеснить родные средства IPC KDE и GNOME новой системе пока не удалось [правда, в KDE 4 D-Bus все-таки будет использоваться вместо DCOP, – *прим. ред.*], но в процессе разработки D-Bus обрела несколько уникальных и полезных особенностей. Важными отличительными чертами D-Bus являются система сигналов и асинхронных вызовов методов, а также система управления выполнением приложений. Таким образом, ответ на вопрос, зачем вам может понадобиться программирование D-Bus, состоит из двух частей. Во-первых, многие важные приложения и системные компоненты (например, Linux HAL и NetworkManager) используют D-Bus как средство общения с внешним миром. Во вторых, D-Bus – это платформенно-независимая система IPC, которая присутствует практически в каждом дистрибутиве Linux и устанавливается по умолчанию во многих из них. Поэтому, если вы пишете приложение, которое должно предоставлять сервисы IPC, не являясь частью какого-либо рабочего стола, вам, безусловно, имеет смысл обратить внимание на D-Bus. При этом следует учитывать и минусы D-Bus. В системе все еще не реализована связь между разными машинами, хотя работа в этом направлении ведется. D-Bus легко может быть перенесена на другие Unix-платформы, но ее версия для Windows по-прежнему далека от завершения.

Среди конкурирующих технологий (в том смысле, что их зачастую можно использовать вместо D-Bus), следует отметить CORBA, SOAP, XML-RPC, DCOM, DCOP, Bonobo. Чем D-Bus отличается от них? CORBA, как и D-Bus, использует быстрый бинарный протокол. В отличие от D-Bus, CORBA предназначается для решения чрезвычайно широкого круга задач и может использоваться как в локальной, так и в распределенной системе. В CORBA отсутствуют такие элементы D-Bus, как система управления выполнением приложений и система сигналов. SOAP и XML-RPC представляют собой протоколы, в которых на низком уровне активно используется XML. Эти технологии межпроцессного взаимодействия хорошо подходят для Интернета, однако при обмене данными между приложениями, работающими на одной машине, использование механизмов XML приводит к пустой трате ресурсов (при этом надо отметить, конечно, что приложения, использующие эти протоколы, чрезвычайно легко масштабируются). Технологии DCOM, DCOP и Bonobo имеют один сходный недостаток – каждая из них предназначена для конкретной платформы (Windows, KDE и GNOME соответственно), и организовать взаимодействие между приложениями разных платформ с их помощью будет весьма непросто.

Немного об архитектуре

В основе структуры D-Bus лежит понятие шины (bus). Шина представляет собой механизм, с помощью которого процессы обмениваются данными. Хотя, в принципе, любые два процесса могут организовать «частную» шину средствами D-Bus и обмениваться данными между собой, интерес представляют общедоступные шины, которые поддерживает демон D-Bus. Исполняемый файл демона имеет имя dbus-daemon. Обычно, если демон D-Bus приходится запускать вручную, используется команда dbus-launch.

Демон D-Bus предоставляет нам две шины: системную шину (system bus) и пользовательскую шину (session-bus). Системная шина может использоваться для передачи данных в масштабах системы, в то время как пользовательская шина позволяет передавать данные между процессами, принадлежащими одному пользователю. Следует учесть, что D-Bus следит за правами пользователей в системе и не позволит вам нарушить политику безопасности Linux с помощью системной шины.

Все процессы, использующие D-Bus для обмена данными, выступают в роли клиентов, которые подключаются к демону D-Bus и, таким образом, получают доступ к одной из шин. Об этом необходимо помнить, между прочим, и для того, чтобы не запутаться в терминологии. Подключаясь к шине, каждый процесс создает соединение (с демоном D-Bus). У каждого соединения есть имя (которое в оригинальной литературе обозначается терминами connection name и bus name). Имена соединений похожи на имена интернет-узлов, вывернутые наизнанку. Например, менеджер HAL создает соединение с именем org.freedesktop.Hal, а клиент *Skype* – с именем com.Skype.API. Поскольку все прило-



жения, использующие системную или пользовательскую шины D-Bus, соединяются с демоном D-Bus, а не друг с другом, существует возможность использовать одно соединение D-Bus для обмена данными между разными приложениями.

Непосредственно процесс обмена реализован в форме отправки сообщений D-Bus, которые, в свою очередь, делятся на три категории: запросы (requests), ответы на запросы и сигналы. Сообщение-запрос несет адресату некоторую информацию и предполагает получение ответного сообщения. Сообщение-сигнал не предполагает ответа. Системы сообщений «запрос-ответ» формируют двусторонние соединения, тогда как сигналы допускают широковещательную рассылку (в этом случае соответствующее сообщение получает каждый процесс, зарегистрировавшийся как получатель сигнала).

Каждое приложение, которое предоставляет другим приложениям некие сервисы D-Bus, формирует объекты D-Bus, которые являются адресатами сообщений-запросов (а, значит, источниками сообщений-ответов) и сигналов. Каждое соединение может предоставлять несколько объектов. Вполне естественно, что и у объектов тоже есть имена (их иногда называют «путями объектов»). В отличие от имен соединений, они используют символы косой черты (/), откуда и происходит их второе название. Например, клиент *Skype* создает объект с именем /com/Skype.

Если мы поднимемся выше на один уровень абстракции, то увидим, в чем заключается сходство между объектами D-Bus и объектами ООП. Обмен сообщениями в модели «запрос-ответ» можно рассматривать как вызов метода объекта, в котором сообщение-запрос передает параметры метода, а сообщение-ответ – возвращаемые значения. Именно семантика вызова методов используется при формировании сообщений-запросов и получении ответов D-Bus.

Поскольку в основе вызовов методов объектов D-Bus лежит обмен сообщениями, существует возможность асинхронного вызова. Вызвав метод объекта D-Bus, программа может выполнить какие-то операции, не дожидаясь получения ответа. Можно даже вызвать еще один метод объекта до того, как был получен результат предыдущего вызова. Сообщения-сигналы проще всего сравнить с сигналами *Qt*.

Совокупность методов и сигналов, предоставляемых объектом D-Bus, называется интерфейсом объекта D-Bus. Декларируя поддержку определенного интерфейса, объект гарантирует определенную реакцию на сообщения, поступающие извне. Каждый объект может поддерживать несколько интерфейсов, которые, естественно, различаются по именам. Некоторую путаницу создает тот факт, что имена интерфейсов имеют ту же структуру, что и имена соединений (и часто совпадают с ними). Например, объект /com/Skype экспортирует интерфейс com.Skype.API (такое же имя имеет соединение, которое создает клиент *Skype*). Итак, для передачи сообщения другой программе с помощью D-Bus, необходимо знать имя соединения этой программы с демоном D-Bus, имя объекта и имя интерфейса. Для вызова метода, декларируемого интерфейсом, нужно, конечно, знать еще и имя метода. Вы можете придти к выводу, что для того, чтобы обмениваться сообщениями D-Bus с другой программой, вам всегда необходимо создавать объект D-Bus и декларировать поддержку некоторого интерфейса. На самом деле, это не обязательно. Если ваша программа использует D-Bus только для вызова методов объектов других приложений, никаких специальных действий по регистрации объектов выполнять не требуется. Более того, вам не придется придумывать для вашей программы и имя соединения – в процессе подключения к шине демон D-Bus создаст его автоматически. Сгенерированное имя будет иметь вид :x.y, где x и y – некоторые числа. На самом деле, подобное имя создается для каждого соединения D-Bus в рамках данной сессии, и именно оно используется демоном для передачи сообщений. В случае с программами, зарегистрировавшими сервисы D-Bus, имя, начинающееся с двоеточия, является синонимом постоянного имени (такого как com.Skype.API) в рамках текущей сессии. Если рассматривать программу, предоставляющую сервис D-Bus, как аналог web-сервера, то имя соединения, начинающееся с двоеточия, можно сравнить с IP-адресом сервера, а постоянное имя – с доменным именем. Иерархия различных элементов D-BUS представлена на **Рис. 1**.

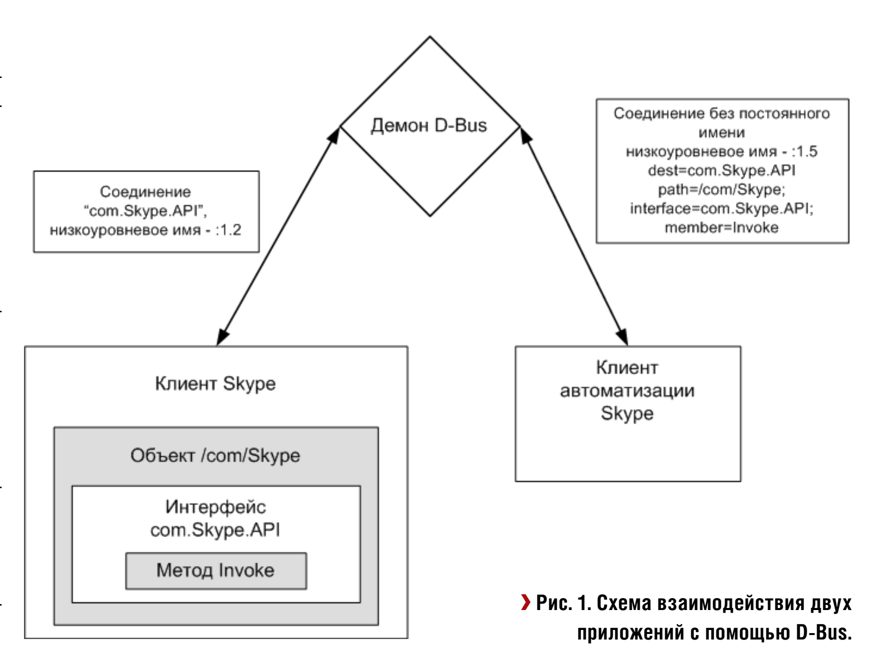


Рис. 1. Схема взаимодействия двух приложений с помощью D-Bus.

Хотя вы и не можете работать с объектами D-Bus напрямую, система предоставляет программистам объектоподобный интерфейс, который реализуется с помощью так называемых прокси-объектов (proxy objects). Прокси можно считать представителем объекта D-Bus в вашей программе. Насколько прокси-объект похож на «настоящий» – зависит от реализации. В языках Java и Python работа с прокси осуществляется практически так же, как с «настоящими» объектами языка. При использовании интерфейсов библиотеки *GLib* для работы с прокси применяется специальный набор функций.

Обращаясь к объектам D-Bus какого-либо приложения, вы предполагаете, что хотя бы один экземпляр этого приложения запущен в системе. А что, если это не так? Выше отмечалось, что система D-Bus способна управлять выполнением приложения. Демон D-Bus может запустить приложение по вашему требованию (для этого, конечно, данное приложение должно быть особым образом зарегистрировано в системе). Этот механизм известен под именем D-Bus Activation.

Подключайтесь!

Основу низкоуровневого D-Bus API составляют два объекта – DBusConnection и DBusMessage. Первый объект инкапсулирует все, что связано с управлением шиной D-Bus, второй позволяет управлять сообщениями. Еще раз напомню, что когда мы говорим об объектах D-Bus API, речь идет не об объектах в смысле ООП, а об объектах в стиле *GTK+* API (интерфейс программирования D-Bus вообще очень похож на интерфейс программирования *GTK+*).

Следующий код представляет собой минимальную программу, использующую возможности D-BUS.

```
#include <stdio.h>
#include <dbus/dbus.h>
int main (int argc, char **argv)
{
 DBusConnection * connection;
 DBusError error;
 DBusMessage *call;
 DBusMessage *reply;
 const char * arg = "PROTOCOL 6\n"
 char * response = NULL;
 dbus_error_init(&error);
 connection = dbus_bus_get(DBUS_BUS_SESSION, &error);
 if (!connection) {
  printf("Ошибка соединения с D-BUS: %s\n", error.message);
```



»

D-Bus интерфейс клиента Skype

Вы уже заметили, что в качестве примера приложения, предоставляющего сервисы D-Bus, мы упоминаем клиент *Skype*. Интерфейс, экспортируемый клиентом *Skype* очень прост, и в то же время демонстрирует все основные возможности D-Bus. Объект /com/Skype поддерживает один-единственный метод – Invoke, позволяющий внешнему приложению передавать команды клиенту *Skype*. Единственным аргументом метода Invoke является строка команды, а возвращаемым значением – строка, в которой содержится ответ программы на переданную команду. Однако клиент *Skype* может не только выполнять команды сторонних приложений, но и передавать им различную информацию, например, о подключении нового пользователя. Чтобы получать сообщения от клиента *Skype*, приложение должно зарегистрировать класс /com/Skype/Client. Когда клиент *Skype* хочет проинформировать приложение о чем-либо, он вызывает метод Notify класса /com/Skype/Client, передавая в единственном параметре этого метода строку-сообщение. Метод Notify не возвращает значений.

»

```
  dbus_error_free(&error);
  return 1;
 }
 call = dbus_message_new_method_call("com.Skype.API", "/com/
Skype", "com.Skype.API", "Invoke");
 dbus_message_append_args (call, DBUS_TYPE_STRING, &arg, DBUS_
TYPE_INVALID);
 reply = dbus_connection_send_with_reply_and_block (connection, call,
100000, &error);
 if (!reply) {
  printf("Ошибка вызова метода: %s\n", error.message);
  dbus_error_free(&error);
  return 1;
 }
 dbus_message_get_args (reply, &error, DBUS_TYPE_STRING,
&response, DBUS_TYPE_INVALID);
 if (response != NULL)
  printf("Ответ: %s\n", response);
 dbus_message_unref(call);
 dbus_message_unref(reply);
 dbus_connection_unref(connection);
 return 0;
}
```

Объявления типов и функций, связанные с D-Bus API, становятся доступны программе при включении в ее текст заголовочного файла **dbus/dbus.h** (где найти этот файл, знает команда *pkg-config --cflags dbus-1*). Соединение с сервером D-Bus устанавливается с помощью функции dbus_bus_get(). Ее первый параметр указывает, к какой шине, системной или пользовательской, мы хотим подключиться (в данном случае выбирается пользовательская шина – DBUS_BUS_SESSION). Вторым параметром должен быть указатель на переменную типа DBusError. Если в процессе подключения к шине произойдет сбой, функция dbus_bus_get() вернет значение **NULL**, а в переменной error будет записана информация об ошибке.

После того как соединение с шиной установлено, мы вызываем метод объекта другого приложения. Вызов метода состоит из четырех этапов: создания сообщения-запроса, создания списка аргументов для вызываемого метода, передачи сообщения и обработки результата вызова.

Сообщение-запрос на вызов метода создается функцией dbus_message_new_method_call(). Ее четырьмя аргументами являются имена соединения удаленного приложения, объекта, интерфейса и вызываемого метода соответственно. Функция возвращает указатель на созданный ею объект DBusMessage, который содержит информацию о новом вызове. Поскольку создаваемое сообщение предназначено для вызова метода, мы должны сформировать список его аргументов. Это делается с помощью функции dbus_message_append_args(). Первый аргумент этой функции – указатель на объект DBusMessage. Далее следует переменное число параметров, которые передают аргументы вызываемого метода. Каждому аргументу соответствуют два параметра функции dbus_message_append_args(). В первом параметре передается константа, указывающая тип аргумента, во втором – указатель на область памяти, в которой хранится его значение. Завершается список аргументов константой DBUS_TYPE_INVALID. Поскольку у вызываемого нами метода Invoke один параметр, мы передаем dbus_message_append_args() список из трех аргументов. Аргумент DBUS_TYPE_STRING указывает тип параметра Invoke, затем следует указатель на значение (в нашем случае – указатель на переменную типа char*), далее – маркер конца списка DBUS_TYPE_INVALID. Отметим, что функция dbus_message_append_args() – не единственное средство создания списка аргументов. Низкоуровневый интерфейс D-Bus предоставляет в наше распоряжение и другие функции, способные формировать списки аргументов динамически, во время выполнения программы.

Теперь наше сообщение-запрос готово к отправке. Как уже отмечалось, существует два варианта вызова методов объектов D-Bus. При асинхронном вызове наша программа продолжит выполнение, не дожидаясь ответа (он будет обработан, когда поступит). При блокирующем вызове работа программы будет приостановлена до тех пор, пока мы не получим ответ (или пока не истечет заданное нами время ожидания). Ради простоты мы воспользуемся блокирующим вызовом. Функция dbus_connection_send_with_reply_and_block() делает именно то, что предполагает ее название – посылает сообщение-запрос и блокирует выполнение программы в ожидании ответа. Первый аргумент функции – указатель на объект DBusConnection. Далее следует указатель на объект-сообщение. Затем мы указываем максимальный интервал ожидания ответа (в миллисекундах). Последний аргумент dbus_connection_send_with_reply_and_block() – адрес переменной DBusError. В случае успешного завершения функция возвращает указатель на объект DBusMessage, который содержит сообщение-ответ. В нашем примере это сообщение передает нам значение, возвращенное методом Invoke. Тут следует внести небольшое уточнение. В процессе вызова dbus_connection_send_with_reply_and_block() может возникнуть две разновидности ошибок. Ошибки первой разновидности связаны с возможными сбоями при передаче сообщения-запроса или ответа на него. При возникновении такой ошибки функция dbus_connection_send_with_reply_and_block() возвращает значение **NULL**, а описание ошибки следует извлекать из переменной error (то же самое происходит, если превышен интервал ожидания ответа). Но возможно возникновение ошибки и другого рода, при которой функция dbus_connection_send_with_reply_and_block() вернет указатель на объект DBusMessage, как и в случае нормального завершения вызова, только соответствующее сообщение будет не ответом на вызов метода, а сообщением об ошибке. Наша программа не обрабатывает возможные ошибки при вызове методов (да и вызвать ошибку в методе Invoke не так-то просто).

Получив ответное сообщение и сохранив указатель на него в переменной reply, мы извлекаем значение, возвращенное методом Invoke. Делается это с помощью функции dbus_message_get_args(). Первый аргумент функции – указатель на объект-сообщение. Далее следует указать на переменную типа DBusError, в которой будет сохранена информация об ошибке, если таковая возникнет. Затем – тот же список аргументов переменной длины, что и у dbus_message_append_args(): тип параметра, переменная для сохранения адреса его значения и маркер конца списка. Важное отличие dbus_message_get_args() от dbus_message_append_args() заключается, конечно, в том, что теперь мы не передаем, а получаем значения параметров. В переменную response в нашем примере записывается адрес строки, возвращенной методом Invoke. Сама строка хранится в недрах сообщения reply, и мы не должны пытаться высвободить (или модифицировать) занимаемую ею память.

На этом работа нашей программы окончена. С помощью функций dbus_message_unref() и dbus_connection_unref() мы сообщаем системе, что созданные нами объекты интерфейса D-Bus нам больше не нужны, и выделенную под них память можно освободить.

А дальше?

Сценарий, в котором программа вызывает метод объекта, экспортируемого другим приложением, выглядит довольно просто. Однако все



становится гораздо сложнее, если мы хотим предоставить удаленному приложению возможность вызывать методы объекта D-Bus, экспортируемого нашим приложением. Если вы думаете, что при работе с D-Bus вы сможете избежать создания собственных объектов, то вы ошибаетесь. Возьмем для примера тот же *Skype* API. После того как приложение установит связь с клиентом *Skype*, клиент время от времени вызывает метод Notify объекта /com/Skype/Client нашего приложения. Точнее говоря, пытается вызвать. Если объекта не существует (как в приведенном выше примере), ничего катастрофического не случится. Мы просто не получим сообщения от клиента *Skype*. Сама регистрация объекта – дело несложное (для этого используется функция dbus_connection_register_object_path()). Сложность заключается в том, что, зарегистрировав собственный объект, программа должна быть готова обрабатывать сообщения D-Bus, посылаемые ей системой. В простом примере, приведенном выше, все сообщения генерировались по нашей инициативе. Для обработки сообщений, поступающих асинхронно, нам потребуется создать цикл обработки сообщений D-Bus. Это можно сделать многими способами (на моем сайте, по адресу **symmetrica.net/d-bus-demo-1**, вы найдете пример создания цикла обработки сообщений низкоуровневыми средствами).

Думаю, вы уже поняли, что работать с D-Bus с помощью низкоуровневого API не очень удобно. Неудивительно, что программисты создали многочисленные привязки D-Bus API к различным языкам программирования и библиотекам. В настоящее время D-Bus поддерживается в *GTK+/GLib* (следует отметить, что это – наиболее проработанные привязки), *Qt 3/Qt 4*, Python, Java, Perl. Сам я работаю над привязками D-Bus для *wxWidgets*.

Привязки D-Bus решают три задачи. Во-первых, выполняется интеграция цикла обработки сообщений D-Bus и целевой платформы. Во-вторых, объектная модель D-Bus API отображается в объектную модель, принятую на целевой платформе. В третьих, создаются методы для работы с D-Bus прокси, как с «родными» объектами. Но все это уже совсем другая история... LXF







ЧАСТЬ 2 Сегодня **Андрей Боровский** разберется с вопросами, без которых немыслимо создание любого приложения *wxWidgets*: расположением виджетов в окне, обработкой сообщений и интернационализацией.

Сегодня мы рассмотрим основные моменты программирования с *wxWidgets*: компоновку дочерних визуальных элементов, обработку событий и управление потоками на примере приложения, взаимодействующего с клиентом *Skype*. В результате получится программа *Skype Monitor*, представленная на **Рис. 1**. Эта утилита воспроизводит функциональность одного из демо-приложений, написанных программистами Skype Limited на *Qt*. Наш вариант использует, естественно, *wxWidgets*. Программа позволяет посылать команды клиенту *Skype*, используя Skype Public API, и получать ответные сообщения. С помощью команд *Skype Monitor* можно даже записывать звонки *Skype*, хотя процесс и не автоматизирован. Базовая структура программы *wxWidgets* нам уже знакома, но вкратце напомню: мы создаем класс MyApp, производный от wxApp, и переопределяем в нем метод OnInit() класса-предка. Мы также создаем класс MainFrame, производный от wxFrame. Объект этого класса, создаваемый в методе MyApp::OnInit(), реализует главное окно программы.

```
Skype Monitor
PROTOCOL 6                                Send
COMMAND: NAME wxWidgets_SKYPE_Monitor
RESPONSE: OK
COMMAND: PROTOCOL 6
RESPONSE: PROTOCOL 6
NOTIFICATION: CONNSTATUS ONLINE
NOTIFICATION: CURRENTUSERHANDLE ksh.linker
NOTIFICATION: USERSTATUS ONLINE
```

› **Рис. 1. Программа *Skype Monitor*.**

Расположение дочерних элементов

Расположение дочерних визуальных элементов в *wxWidgets* контролируют специальные объекты – «sizers». По аналогии с *Qt*, я буду называть их менеджерами компоновки. Термин «контейнеры» был бы, наверное, уместнее, но контейнерами в *wxWidgets* именуются окна специальных типов (например, wxPanel). Менеджеры компоновки *wxWidgets* напоминают менеджеры компоновки *Qt* и контейнеры *GTK+*. От базового класса wxSizer происходит несколько специальных (wxBoxSizer, wxGridSizer, wxFlexGridSizer, wxStaticBoxSizer). Каждый из них реализует одну из простых схем расположения визуальных элементов. Например, wxBoxSizer может располагать дочерние элементы либо вертикально, в столбце, либо горизонтально – в линейке. Для конструирования более сложных интерфейсов создаются иерархии из менеджеров.

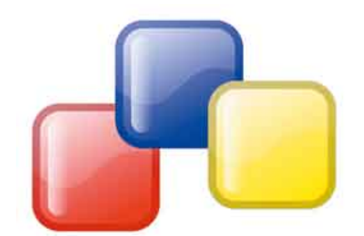
Наше приложение использует для построения интерфейса два объекта wxBoxSizer (переменные vbox и hbox). Объект vbox нужен нам для того, чтобы расположить в один ряд строку ввода команды и кнопку Send. Таким образом, создается единый блок «строка ввода + кнопка». Сам объект vbox становится дочерним элементом объекта hbox, который располагает свои дочерние элементы вертикально. Вторым дочерним элементом hbox становится окно просмотра результатов команды. Когда мы говорим о дочерних элементах объектов vbox и hbox, мы должны помнить, что класс wxSizer и его потомки не являются потомками wxWindow (в этом они похожи на менеджеры компоновки *Qt*). Каждый дочерний элемент объектов hbox и vbox является также и дочерним элементом главного окна.

Давайте посмотрим, как все это выглядит в коде. Ниже приводится фрагмент из конструктора класса MainForm:

```
vbox = new wxBoxSizer(wxVERTICAL);
hbox = new wxBoxSizer(wxHORIZONTAL);
commandEdit = new wxTextCtrl(this, ID_ENTER, "PROTOCOL 6",
wxDefaultPosition, wxDefaultSize, wxTE_PROCESS_ENTER);
hbox->Add(commandEdit, 1, wxALL, 2);
commandButton = new wxButton(this, ID_START, "Send");
hbox->Add(commandButton, 0, wxALIGN_LEFT|wxALL, 2);
vbox->Add(hbox, 0, wxEXPAND);
logViewer = new wxTextCtrl(this, wxID_ANY, "", wxDefaultPosition,
wxDefaultSize, wxTE_MULTILINE|wxTE_READONLY|wxTE_RICH);
vbox->Add(logViewer, 1, wxEXPAND|wxALIGN_CENTER|wxALL, 2);
SetSizer(vbox);
vbox->Fit(this);
```

Создавая объект vbox, мы передаем в параметре конструктора константу wxVERTICAL, которая указывает, что новый менеджер компоновки должен располагать дочерние элементы вертикально. Аналогично создается объект hbox, только на этот раз мы задаем горизонтальное расположение дочерних элементов. Объект commandEdit класса wxTextCtrl реализует строку ввода команды. После того, как мы создали этот объект, мы добавляем его в менеджер компоновки hbox с помощью метода Add(). Точно так же мы добавляем в менеджер компоновки объект commandButton, представляющий кнопку. Обратите внимание на списки аргументов метода Add() в первом и во втором случаях. Первый параметр, естественно, указатель на добавляемый объект. Далее следует аргумент, который можно уподобить параметру stretch factor менеджеров компоновки *Qt*. Он определяет, должны ли размеры визуального элемента меняться вместе с размерами окна.

» **Месяц назад** Мы познакомились с *wxWidgets* и ее базовыми инструментами для разработки.



И компоновка

Следующий аргумент представляет собой комбинацию флагов, управляющих изменением размера добавляемого элемента, его выравниванием и применением отступов. Константа wxEXPAND указывает на то, что добавляемый элемент должен заполнять все доступное свободное пространство менеджера компоновки. Константа wxLEFT выравнивает соответствующий визуальный элемент по левому краю. Константа wxALL определяет, что отступ (ширина которого указывается в следующем параметре) должен быть применен по всем краям дочернего элемента. Мы делаем менеджер компоновки hbox дочерним элементом vbox, в результате чего vbox управляет строкой ввода команд и кнопкой как единым элементом. Вторым дочерним элементом vbox становится объект logViewer (окно просмотра результатов команд).

Вернемся к изменению размеров дочерних элементов. Фактически, этим процессом управляют значения второго и третьего параметров метода Add(). Второй параметр указывает, должны ли размеры дочернего элемента меняться в «главном» направлении менеджера компоновки wxBoxSizer (то есть, для «вертикальных» менеджеров – по высоте, а для «горизонтальных» – по ширине). Что касается изменения размеров элемента в другом направлении (например, при изменении ширины «вертикального» менеджера), то на него влияет наличие константы wxEXPAND в третьем параметре, а также свойства самого дочернего элемента. Мы подбираем значения этих параметров, исходя из соображений здравого смысла. Кнопка должна иметь фиксированную ширину и высоту, поэтому во втором параметре Add() передается значение **0**, а в третьем отсутствует константа wxEXPAND. Строка ввода должна быть фиксирована по высоте, но ее ширина может меняться при изменении ширины окна. Соответственно, во втором параметре Add() передается значение **1**. Мы нигде не указываем специально, что высота строки ввода и кнопки должна быть постоянна, но этого и не требуется, так как высота этих элементов фиксирована по умолчанию. Для того, чтобы высота объекта hbox не менялась, мы передаем значение **0** во втором параметре вызова Add(), который добавляет этот объект в объект vbox. Окно logViewer, которое представляет собой объект того же класса wxTextCtrl, но только в многострочном варианте, должно иметь переменную высоту и ширину, поэтому во втором параметре метода Add() передается значение **1**, а в третий параметр добавляется константа wxEXPAND.

Нам остается объяснить главному окну, что управление его дочерними элементами доверено объекту vbox и его дочерним элементам. Для этого мы вызываем метод SetSizer() класса wxFrame, имя которого говорит само за себя. Далее мы вызываем метод Fit() нашего «главного» менеджера компоновки vbox и передаем ему указатель на объект главного окна. Метод Fit() подстраивает размеры окна под размеры контейнера (а не наоборот, как можно было бы ожидать).

События

Как уже отмечалось, *wxWidgets* поддерживает два механизма обработки событий. Более старый механизм, использующий статические таблицы обработчиков событий, был навеян библиотекой MFC. Новый вариант, основанный на классах-обработчиках, позволяет назначать обработчики динамически, во время выполнения программы. Этот механизм напоминает сигналы и слоты, используемые в *Qt*. В книге «*Cross-Platform GUI Programming with wxWidgets*» (напоминаю адрес: **phptr.com/perens**) внимание читателя (и авторов) сосредоточено, в основном, на обработке событий в стиле MFC. В противоположность этому, мы углубимся в обработку событий «в стиле *Qt*» и лишь кратко рассмотрим альтернативный вариант.

Система обработки событий *wxWidgets* основана на классе wxEvtHandler. Этот класс используется и при обработке событий, обработчики которых заданы в таблицах, и при обработке событий «в стиле *Qt*». Для динамического назначения и удаления обработчика служат методы Connect() и Disconnect() класса wxEvtHandler. Класс wxEvtHandler является одним из предков базового для всех окон класса wxWindow, так что внутри методов потомков wxWindow методы Connect() и Disconnect() доступны всегда. В конструкторе класса MainFrame мы, помимо прочего, назначаем обработчик событию wxEVT_COMMAND_BUTTON_CLICKED кнопки commandButton. Для этого используется следующий вызов Connect():

```
Connect(ID_START, wxEVT_COMMAND_BUTTON_CLICKED,
wxCommandEventHandler(MainFrame::OnCommand));
```

Первый параметр метода Connect() – это идентификатор кнопки, который, как мы видели выше, был передан ей в конструкторе. В *Qt* и *GTK+* в этом месте можно было бы ожидать использование адреса объекта-кнопки. Практика использования назначенных объектам числовых идентификаторов заимствована в *wxWidgets* из MFC. Библиотека предоставляет нам ряд идентификаторов по умолчанию. Например, если бы наше окно содержало кнопки OK и Cancel, мы могли бы использовать для них стандартные идентификаторы wxID_OK и wxID_CANCEL, соответственно. Конечно, ничто не мешает нам использовать эти идентификаторы и для других типов кнопок (если их не приходится использовать по назначению), но для того, чтобы код приложения был более понятным, для нестандартных элементов управления лучше использовать нестандартные идентификаторы. Порядок определения собственных идентификаторов элементов управления так же позаимствован из MFC. В *wxWidgets* определена константа wxID_HIGHEST, которая содержит максимальное значение, задействованное библиотекой в качестве идентификатора. Мы можем использовать для собственных идентификаторов любые значения, превышающие wxID_HIGHEST, не опасаясь, что они совпадут с уже определенными идентификаторами. Например, константа ID_START определена нами как

```
#define ID_START wxID_HIGHEST + 1
```

Если нам понадобятся еще идентификаторы, мы воспользуемся значениями wxID_HIGHEST + 2 и т.д.

Вторым аргументом метода Connect() является имя события. В третьем параметре методу передается указатель на обработчик события. В объявлении метода Connect() третий параметр имеет тип wxObjectEventFunction. Указатель на фактически используемый метод-обработчик должен быть приведен к этому типу с помощью специального макроса. В нашем примере обработку события выполняет метод MainFrame::OnCommand(), а необходимое преобразование типа выполняется с помощью макроса wxCommandEventHandler(). Этот макрос используется для всех событий, чьи имена имеют префикс wxEVT_COMMAND (события, которые генерируются различными элементами управления пользовательского интерфейса). У метода Connect() есть и другие параметры, но их значения редко задаются явным образом.

Определение метода-обработчика OnCommand выглядит так:

```
void MainFrame::OnCommand(wxCommandEvent& event)
```

Обработчик не возвращает значений, а единственным его аргумен-

том является ссылка на объект класса wxCommandEvent. Один и тот же обработчик может использоваться для обработки самых разных событий от разных элементов управления. В этом случае из параметра event мы можем получить дополнительную информацию о событии. Класс wxCommandEvent является потомком класса wxEvent – базового для аргументов всех обработчиков событий.

Для генерации событий класс wxEvtHandler и его потомки предоставляют нам методы ProcessEvent() и AddPendingEvent(). Первый метод непосредственно вызывает обработчик события, второй метод позволяет поместить событие в очередь на обработку. У обоих методов определен только один параметр, в котором передается ссылка на объект класса-потомка класса wxEvent. Основываясь на типе этого объекта, *wxWidgets* определяет, обработчик какого события следует вызвать. Если с методом ProcessEvent() вам, скорее всего, не придется иметь дела, если только вы не пишете собственный визуальный элемент для *wxWidgets*, то метод AddPendingEvent() используется довольно часто, и мы к нему еще вернемся.

Итак, подведем итог. При создании каждого визуального элемента, который может быть источником событий, мы присваиваем ему числовой идентификатор, заданный константой (напомню, что мы можем выбрать значение wxID_ANY, если нам не нужно обрабатывать события данного элемента). Чтобы назначить обработчик события, мы вызываем метод Connect() объекта, реализующего главное окно. Этому методу передаются идентификатор визуального элемента, имя события и обработчик события. Однако не со всеми компонентами *wxWidgets* дело обстоит так же просто. Рассмотрим, например, класс wxSocketServer, который инкапсулирует сокет, открытый для приема входящих соединений. Каждый раз, когда состояние сокета меняется (например, поступает запрос на соединение), объект этого класса генерирует событие wxEVT_SOCKET. Мы могли бы ожидать, что назначить обработчик этого события так же легко, как и для события wxEVT_COMMAND_BUTTON_CLICKED объекта-кнопки, но это не так. Прежде всего, в конструкторе wxSocketServer нет параметра, в котором можно было передать идентификатор объекта (как это делается при конструировании кнопки), а без идентификатора связать объект с обработчиком события нельзя. Почему в конструкторе wxSocketServer нельзя назначить объекту идентификатор? Идентификаторы обрабатываются объектами, приводимыми к типу wxEvtHandler, например, потомками класса wxWindow. Класс wxSocketServer не является потомком wxEvtHandler, поэтому своего средства обработки сигналов у него нет. Это логично, если вспомнить, что в основе обработки событий лежит механизм обработки сообщений оконной системы, а объект wxSocketServer сам по себе не связан ни с каким окном. Для того, чтобы наладить обработку событий wxSocketServer, нам придется «одолжить» объект, приводимый к wxEvtHandler у какого-либо окна. Сделать это можно, например, так:

```
sockServ = new wxSocketServer(ip);
sockServ->SetEventHandler(*this, ID_SOCKET);
sockServ->SetNotify(wxSOCKET_CONNECTION_FLAG);
sockServ->Notify(TRUE);
Connect(ID_SOCKET, wxEVT_SOCKET, wxSocketEventHandler(MainFrame
::OnConnection));
```

Приведенный фрагмент позаимствован из конструктора некоего класса-потомка *wxFrame*. Переменная sockServ – это указатель на объект wxSocketServer. В конструкторе wxSocketServer передается переменная типа wxSockAddress, которая нас сейчас не интересует. Мы связываем объект wxSocketServer с объектом wxEvtHandler с помощью метода SetEventHandler(). Первым параметром метода должна быть ссылка на объект класса wxEvtHandler (или класса, приводимого к нему). Поскольку этот код вызывается из конструктора главного окна, а класс окна является потомком wxEvtHandler, мы можем передать в этом параметре значение *this. Вторым аргументом метода SetEventHandler() должен быть идентификатор объекта, который используется для назначения обработчика событий (таким образом мы связываем объект sockServ с идентификатором). Константа ID_SOCKET определена нами так же, как и константа ID_START в рассмотренном выше примере. Метод SetNotify() позволяет нам указать, какие именно внутренние события сокета должны приводить к генерации события wxEVT_SOCKET (благодаря этому методу мы можем отсеять неинтересные для нас события еще до вызова обработчика wxEVT_SOCKET). Наконец, метод Notify(), вызванный с параметром TRUE, включает передачу событий объектом-сокетом. Теперь объект sockServ готов к связыванию с обработчиком событий, которое мы выполняем с помощью метода Connect() того же объекта wxEvtHandler, с которым мы связали сокет (то есть объекта this).

Посмотрим теперь, что требуется для генерации события. Допустим, мы хотим сгенерировать событие wxCommandEvent из метода некоторого класса-потомка wxWindow. Прежде всего, нам следует создать объект класса wxCommandEvent:

```
wxCommandEvent event(EVENT_TYPE, CONTROL_ID);
```

Константа EVENT_TYPE обозначает здесь конкретный тип события wxCommandEvent, а константа CONTROL_ID – это тот самый идентификатор объекта-источника события, о котором мы много говорили выше. Для передачи события мы вызываем метод AddPendingEvent():

```
AddPendingEvent(event);
```

Вызов AddPendingEvent() не приводит к немедленному выполнению обработчика события. Он является асинхронным и просто ставит событие в очередь на обработку (поэтому AddPendingEvent() очень удобно использовать тогда, когда нам требуется передать событие главному потоку программы из вспомогательного потока). Когда очередь дойдет до обработки нашего события, соответствующий обработчик получит копию объекта event в качестве аргумента.

Рассмотрим этот механизм внимательней. Мы передаем ссылку на объект события методу AddPendingEvent(), после чего управление событием переходит к *wxWidgets*. Мы не знаем, когда именно будет вызван обработчик события, и не можем контролировать время жизни объекта-события. Дабы избавить нас от хлопот, метод AddPendingEvent() копирует переданный ему объект-событие и использует далее эту копию, а созданный нами объект остается под нашим контролем. У этого факта есть два следствия. Во-первых, у всех классов-потомков wxEvent должен быть реализован механизм «глубокого» копирования (вы должны помнить об этом, когда будете создавать собственные классы объектов-событий), а во-вторых, методу AddPendingEvent() можно передавать ссылку на объект, созданный в стеке вызывающей функции (что мы и сделали в приведенном выше примере).

Теперь скажем несколько слов о статической обработке событий. Для того, чтобы сделать все «в стиле MFC», мы должны объявить и реализовать таблицы событий. Такая таблица может быть объявлена только в классе-потомке класса wxEvtHandler. Чтобы объявить в классе таблицу событий, достаточно добавить в объявление класса макрос DECLARE_EVENT_TABLE(). Например, если у нас есть класс MyFrame, наследующий классу *wxFrame*, объявление таблицы событий может выглядеть так:

```
class MyFrame : public wxFrame
{
public:
...
void OnButtonClick(wxCommandEvent& event);
private:
DECLARE_EVENT_TABLE()
wxButton * m_button;
};
```

Метод OnButtonClick() должен стать обработчиком события EVT_BUTTON кнопки m_button. В файл реализации класса следует добавить макросы

```
BEGIN_EVENT_TABLE(MyFrame, wxFrame)
EVT_BUTTON(ID_BUTTON, MyFrame::OnButtonClick)
END_EVENT_TABLE()
```

Как видите, описание таблицы событий выглядит довольно просто. Между макросами BEGIN_EVENT_TABLE() и END_EVENT_TABLE() мы добавляем по одному макросу для каждого обрабатываемого события. Имена макросов соответствуют типам событий, первым аргументом

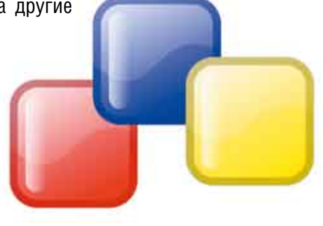
макроса должен быть идентификатор объекта-источника события, вторым – метод-обработчик. Недостатком данного подхода, помимо его громоздкости, является и то, что связывание идентификаторов событий с обработчиками выполняется на этапе компиляции. Кроме того, статические таблицы не могут редактироваться во время выполнения программы (на то они, собственно, и статические).

Интернационализация

При интернационализации приложений *wxWidgets* применяются те же инструменты, что и для других приложений Linux – пакет *gettext* и его друзья, наделенные графическим интерфейсом. Подготовка самих приложений к интернационализации выполняется практически так же, как и при использовании других библиотек. Строки, предназначенные для перевода, выделяются макросом _() или функцией wxGetTranslation(). Любопытно, что строки, не предназначенные для перевода, тоже следует выделять макросами wxT() или _T(), если используется Unicode. Загрузкой ресурсов интернационализации управляет класс wxLocale. Объект класса wxLocale создается обычно в методе OnInit() объекта-потомка wxApp. Далее вызываются, как минимум, два метода объекта wxLocale. Метод Init() включает поддержку выбранной локали и загружает каталог сообщений *wxWidgets*, используемый библиотекой по умолчанию. Метод AddCatalog() используется для загрузки каталога сообщений данного приложения.

В заключение этого краткого обзора *wxWidgets* я хотел бы отметить одну важную, хотя и не очень заметную особенность этого набора виджетов. Код, который создается на *wxWidgets*, обычно изолирован от низкоуровневых элементов системы значительно сильнее, нежели код, использующий *Qt* или *GTK+*. По сравнению с *Qt* и *GTK*, в *wxWidgets* совсем не так просто организовать обработку сообщений X11 или получить, например, дескриптор сокета. С одной стороны это хорошо, так как у программистов не возникает соблазна задействовать специфические для данной системы функции и сделать код приложения менее переносимым. С другой стороны, при попытке расширить возможности *wxWidgets* за счет подключения новых интерфейсов недоступность низкоуровневых функций может вызвать серьезные трудности.

В общем и целом можно сказать, что если для написания «родных» программ Linux лучше подходят *Qt*/KDE и *GTK+*, то библиотека *wxWidgets* может стать подходящим выбором для кроссплатформенных приложений, особенно для таких, которые сначала разрабатываются на платформе Windows, а уже потом переносятся на другие системы. LXF



WavPack:

Не иди на компромисс!

Убеждены, что отношение качества аудиозаписи к размеру содержащего ее файла – величина постоянная?
Александр Супрунов
готов сломать этот стереотип!

С ростом мощностей компьютеров, увеличением емкости жестких дисков и благодаря широкому распространению DVD начали набирать популярность так называемые lossless-кодеки для сжатия звука. Другими словами, уже можно переставать мириться с потерями форматов MPEG Layer 3 или Ogg Vorbis и перейти на следующий уровень, применяя для аудиоданных сжатие без потерь.

На сегодня существует достаточно большое семейство lossless-кодеков: *FLAC* (Free Lossless Audio Codec), *Monkey's Audio*, *WavPack* и даже *WMA9*. Записи, сжатые такими кодеками, имеют больший размер. По крайней мере, так считалось, пока не появился гибридный режим *WavPack*, позволяющий использовать сжатие с потерями, ничего не теряя! О нем сегодня и пойдет речь.

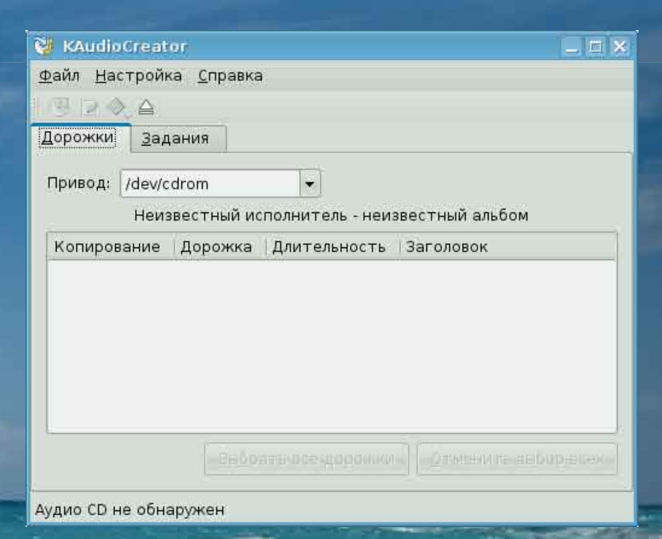
Что это?

WavPack – бесплатный открытый аудиокодек, разработанный Дэвидом Брайантом [David Bryant] и распространяющийся по лицензии BSD. Изначально *WavPack* был сугубо lossless-кодеком, но позднее в нем появилась уникальная возможность, резко выделяющая его из череды других решений – гибридный режим.

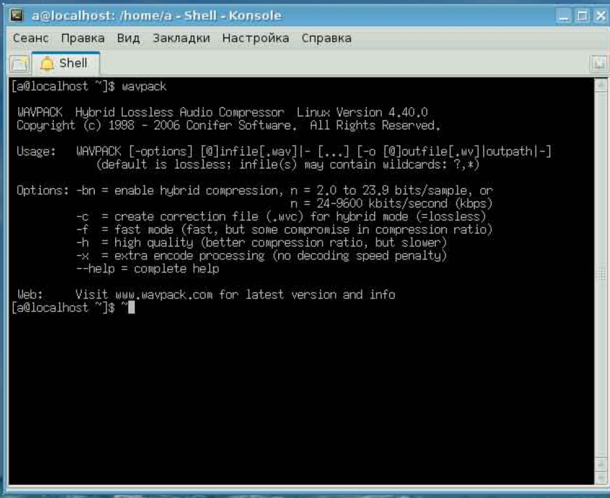
При кодировании в этом режиме создаются два файла – один сжат по принципу lossy (.wv) и несет все преимущества такого сжатия (имеет минимальный размер) и второй – корректирующий (.wvc). Он используется при восстановлении потерянных данных с помощью утилиты *WvUnpack* – что позволяет бескомпромиссно решить вечную проблему выбора между качеством и количеством.

WavPack поддерживает высокие частоты дискретизации и позволяет сжимать 8/16/24/32-битные аудиофайлы. В lossless-режиме *WavPack* работает подобно стандартному архиватору вроде *Zip*, не приводит к деградации записей, как это происходит в форматах MP3 или Ogg и является идеальным способом архивирования аудиоконтента. Степень сжатия зависит от музыкального материала: композиции с более широким динамическим диапазоном сжимаются сильнее. В среднем, при lossless-режиме можно рассчитывать на 50%. В гибридном режиме lossy-файл будет лишь слегка превосходить по размерам файлы вездесущего MP3.





› **Рис. 1. Сохранить содержимое Audio CD в файл можно с помощью *KAudioCreator*.**



› **Рис. 2. Вывод утилиты *wavpack*.**

Прежде чем углубляться в сферу практического использования *WavPack*, давайте рассмотрим основные преимущества кодека. Это, в первую очередь, быстрое сжатие (существует возможность кодировать симметрично/асимметрично, что позволяет влиять на время декомпрессии), открытый исходный код, гибридный режим, поддержка многоканального звука, аппаратная поддержка (iRiver H10/H1xx/H3xx, iAudio X5, Apple iPod, Toshiba Gigabeat), мультиплатформенность, ReplayGain (технология, уравнивающая громкость разных файлов при воспроизведении) и много чего другого. Хотелось бы еще раз подчеркнуть очень быстрое сжатие – в несколько раз более быстрее, чем сжатие в MP3 кодеком *Lame*.

Приступим к делу

Ну что же, давайте попрактикуемся. Вероятно, кодек *WavPack* уже есть в вашем дистрибутиве (в моем ALT Linux 4.0 – точно), если же нет – ищите соответствующие RPM/Deb-пакеты или скачивайте исходные коды. Официальный сайт кодека – www.wavpack.com, но вы также можете найти *WavPack* на нашем DVD.

Сборка из исходных текстов не вызывает сложностей и ограничивается стандартными

./configure (можете указать ключ --enable-mmx для использования MMX-инструкций вашего процессора), make и make install.

Следующий вопрос – откуда взять неискаженный звук? Можно, например, «оцифровать» CD. Этот вопрос я оставлю на самостоятельное изучение, скажу лишь, что для решения задачи существуют как графические (*KAudioCreator*, см. **Рис. 1**), так и консольные утилиты. Важно лишь иметь возможность получить на выходе файл в формате WAV. В дальнейшем я буду предполагать, что он называется **forest.wav**.

Перейдите в консоль и наберите *WavPack*. В окне терминала отобразятся ключи, которые можно использовать при кодировании, и синтаксис команды (**Рис. 2**):

```
wavpack [-options] [@]infile[.wav]|- [...] [-o [@]outfile[.wv]|
```

В простейшем случае достаточно указать оригинальный файл и название выходного файла в формате *WavPack* (**.wv**), используя ключ -o:

```
wavpack forest.wav -o forest.wv
```

В результате будет создан файл **forest.wv**, сжатый lossless-кодеком. В моем случае его размер уменьшился с 24,1 МБ до 10,9 МБ, и заметьте – без деградации аудиоматериала. Пожелав улучшить (или ухудшить) этот результат, я мог бы воспользоваться следующими ключами:

» -bn Включает гибридный режим. Значение n указывает число бит на сэмпл (от 2.0 до 23.9).

» -c Создает корректирующий файл (**.wvc**) для гибридного режима; -cc обеспечивает максимальную (для гибридного же режима) степень сжатия.

» -f Быстрый режим: повышает скорость работы, понижает степень сжатия.

» -h Опция, обратная -f. Сжимает медленно, но верно. Если же вам и этого мало, используйте -hh: качество и степень сжатия будут очень высокими, но использовать такой файл в портативном проигрывателе не рекомендуется.

» -x Увеличивает скорость декодирования в будущем за счет замедления кодирования в настоящем.

Полный список ключей можно получить, набрав wavpack --help.

Высший пилотаж

Настало время активировать самый интересный, гибридный режим. За него отвечают ключи -b и -c. Если не использовать -c, то корректирующий файл создан не будет и вы получите обычное lossy-сжатие без возможности восстановить оригинал. Например, набрав

```
wavpack -b2.0 forest.wav -o forest.wv
```

вы получите ролик наихудшего возможного качества и отсутствие корректирующего файла. Можете считать это аналогом MP3. В варианте

```
wavpack -b16 forest.wav -o forest.wv
```

качество будет повыше, но и размер выходного файла, соответственно, увеличится.

Но мы, помнится, собрались здесь вовсе не для того, чтобы сжимать аудио lossy-кодеком. Давайте создадим корректирующий файл, способный вернуть оригинальное звучание без потерь. Для это пригодится ключ -c:

```
wavpack -cb16  forest.wav -o forest.wv
```

В результате будет создан выходной файл **forest.wv** и корректирующий файл **forest.wvc**. Оптимальный, на мой взгляд, результат дает строка вида:

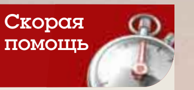
```
wavpack -chhdb16  forest.wav -o forest.wv
```

Она создает корректирующий файл и одновременно удаляет оригинальный. При таком раскладе размер файла уменьшается с 24 МБ до 3,4 МБ (**forest.wv**), а корректирующий файл **forest.wvc** занимает 8,4 МБ.

Но в вашей аудиоколлекции наверняка будет больше одного файла. Чтобы разобраться с ними скопом, используйте команду вида

```
wavpack -chdb2.0 *.wav -o /home/suprunov/music/
```

Остается последний вопрос: на чем прослушать эту новенькую, только что сконвертированную в коллекцию «**.wv** + **.wvc**»? О, здесь изобилие вариантов: поддержка *WavPack* добавлена в *FFmpeg* и *xine-lib*, а значит, доступна через большинство популярных оболочек. Попробуйте *Amarok*, *Audacious*, *Quod Libet*, XMMS... А удовлетворившись программными проигрывателями, присмотритесь к аппаратным. Забудьте MP3, живите в XXI веке, используйте lossless-кодеки, Linux, бластеры и флаеры. Будущее не остановить! LXF



Пока вы еще не успели окончательно запутаться, поясню терминологию: lossy-кодеки сжимают с потерями (при кодировании отбрасывается часть информации, которую вы не в состоянии услышать согласно выбранной психоакустической модели), lossless-кодеки оставляют качество оригинала без изменений.





Ответы

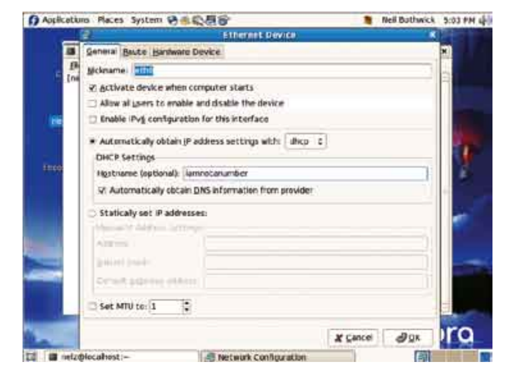
Есть вопрос по Open Source? Пишите нам по адресу: answers@linuxformat.ru

Сегодня мы ответим на вопросы про:

1 Gentoo сменил мне имя

В У меня Fedora Core 7; после того, как попробовал Live CD Gentoo с LXF96, я заметил, что в терминале имя стало root@livecd. Как вернуть его в исходное состояние? Почему такое случилось?
Джон [John]

О Вы, видимо, используете DHCP для конфигурирования сети (возможно, встроенный в маршрутизатор). Подозреваю, что при загрузке Live CD сказал: «меня зовут *livecd*, дайте мне IP-адрес». Затем маршрутизатор запомнил это имя хоста и передал его в следующий раз, когда получил запрос с того же MAC-адреса.



› **Укажите в Настройке сети Fedora имя своего компьютера, даже если используете DHCP.**

В таком случае надо заставить ваш компьютер вместе с DHCP-запросом посылать имя хоста, установив его в конфигураторе сети. Войдите в System > Administration > Network (Система > Администрирование > Сеть), выберите интерфейс и нажмите Edit (Изменить). Введите нужное имя хоста в необязательное поле Hostname. Теперь Fedora будет сообщать DHCP-серверу, что хочет использовать это имя вместо предоставляемого DHCP-сервером.

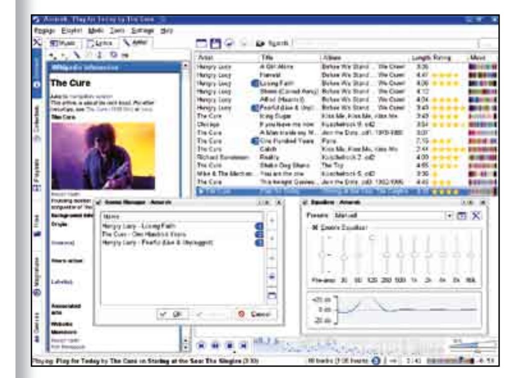
Есть и альтернативный метод, зависящий от вашего маршрутизатора/сервера DHCP. Некоторые из них позволяют указать имя хоста или IP-адрес, выдаваемые конкретному MAC-адресу (MAC-адрес – это аппаратный идентификатор сетевого интерфейса). Если ваш маршрутизатор именно такой, установите имя хоста в нем – он покажет вам список подключенных MAC-адресов, или можно найти его, запустив *ifconfig* на компьютере: MAC-адрес помещается в первой строке вывода интерфейса, обозначен как HWaddr и выглядит похоже на 00:50:56:C0:00:08. Данный метод обеспечит, чтобы данный компьютер всегда имел одно то же имя или IP-адрес, невзирая на загруженную ОС; это может быть и хорошо, и плохо, в зависимости от ваших предпочтений. **НБ**

2 Молчание CD

В Я не могу слушать аудио-CD ни в моей учетной записи пользователя, ни в учетной записи моего партнера. Звук работает – у меня нет проблем с проигрыванием треков, оцифрованных в моей системе. *KsCD* показывает, что аудио-CD играет, но звука нет. Громкость CD в микшере установлена примерно на 90 %. Мастер-канал установлен на 100 %, и ничего не выключено. Я пытался изменить **/etc/fstab** так, чтобы у **/dev/cdrom** были перечислены параметры **user**, **users**, **noauto**.
Джон, из форумов LXF

О Отвечу сперва на Ваш последний комментарий: аудиодиски вообще не монтируются, и **/etc/fstab** тут ни при чем. Большинство дистрибутивов используют автомонтирование через HAL, поэтому CD с данными, как правило, тоже не нужно прописывать в **/etc/fstab**.

По проблеме аудио: есть два пути получения аудиоданных с диска на звуковую карту. Первый – традиционный, с использованием аудиокабеля, втыкаемого »



› ***KsCD* предоставляет основные возможности воспроизведения CD, но лучшим музыкальным проигрывателем для пользователей KDE остается *Amarok*.**

Наши эксперты

» Мы найдем эксперта на любой вопрос! Вы получите ответ на все: от проблем с установкой или модемом до сетевого администрирования; главное – спросить!



Нейл Ботвик
Как экс-владелец ISP и экс-редактор дисков для нашего журнала, Нейл считает, что в Linux он от скуки на все руки.



Майк Сондерс
Майк был одним из создателей прототипа LXF – Linux Answers. Его специальности – программирование, оконные менеджеры, скрипты инициализации и SNES.



Пол Хадсон
Пол – местный суперпрограммист. Он справится со всеми вашими проблемами по части web и баз данных.



Ник Вейч
В свободное от исчеркивания текстов красными чернилами время Ник возится с Linux- графикой и 3D-приложениями; он у нас отвечает за простые вопросы!



Владимир Иванов
Работает с Linux уже более 10 лет, днем – создавая комплексные решения для крупных компаний, а вечером – модерируя сайт linux.org.ru и отвечая на вопросы о системном администрировании и информационной безопасности.



Валентин Синицын
В редкие свободные минуты главный редактор нашего журнала обычно запускает *mcedit*, чтобы отшлифовать какое-нибудь открытое приложение. Его любимая тема – настольный Linux.

КУДА ПОСЫЛАТЬ ВОПРОСЫ:
Пишите нам по адресу: answers@linuxformat.ru или спрашивайте на форуме: www.linuxforum.ru

» в небольшой плоский четырехполюсный разъем на задней стороне привода и в соответствующий разъем на звуковой карте (или плате, если у вас встроенный звук). Это самый эффективный метод, потому что компьютеру надо только посылать команды **старт/стоп** и т.д. на привод; звук просто проходит по кабелю, где микшируется и выводится звуковой картой.

Второй метод использует цифровое извлечение, когда компьютер снимает данные CDDA (Compact Disc Digital Audio) с диска через ATA/SATA кабели, выполняет необходимые преобразования аудио и передает результат на звуковую карту. Метод требует более интенсивной загрузки процессора, но на современной технике она незаметна, и, как правило, этот метод ставится по умолчанию, поскольку позволяет изготовителю сэкономить пару копеек на кабелях. Существует простой способ проверить, считывает ли CD-привод аудио напрямую: при наличии на его панели разъема для наушников, подключите их туда. Если слышна музыка, то у вас прямое соединение через кабель.

KsCD не использует цифровое извлечение по умолчанию; оно включается установкой флажка Use Direct Digital Playback (Цифровое Воспроизведение) в окне настройки. Вы можете предпочесть другой мультимедийный проигрыватель: *KsCD* сгодится как простой CD-плейер, но если Вы также слушаете аудиофайлы MP3 или Ogg Vorbis или интернет-радио, куда лучше будет *Amarok*, аудиоплейер класса «все включено» для KDE.

3 Перекрасим консоль

Как мне изменить цвет текста и фона на консоли Linux при запуске уровня 3? Я хотел бы иметь черный текст на белом фоне. Если такое возможно, хорошо бы данное изменение применялось уже при загрузке ядра, и все загрузочные сообщения выводились бы черным по белому. Подозреваю, что придется малость раскурочить ядро, а если так, то какую часть исходного кода ядра надо менять?

Гордон [Gordon]

Изменение цвета консоли (после того, как она будет запущена) можно выполнить двумя способами. Вы можете просто вывести коды ANSI, если помните, как они выглядят, или использовать команду *setterm*. Чтобы установить черный на белом цвет, наберите

```
setterm -background white -foreground white store.
```

В man-странице *setterm* содержится полная информация о различных опциях и допустимые значения цветов. Эту команду можно вызвать из стартового скрипта; детали зависят от дистрибутива, но обычно **/etc/rc.local** будет хорошей отправной точкой.

Если Вы хотите других цветов с самого начала, отредактируйте исходники ядра, поменяв цвета, и перекомпилируйте ядро. Нужно изменить файл **drivers/char/vt.c** (в старых ядрах – **drivers/char/console.c**); поищите строки, начинающиеся с

```
vc->vc_def_color        = 0x07; /* white */
```

В исходниках ядра **2.6.22** это строка **2739**. Две шестнадцатеричные цифры означают цвета фона и текста: **0x07**, по умолчанию, белое на черном, а, наоборот, **0x70** – та самая настройка, которая Вам нужна. Измените значения по умолчанию, как Вам нравится, и пересоберите ядро обычным образом. Номера цветов:

- **0** .... **черный**
- **1** .... **синий**
- **2** .... **зеленый**
- **3** .... **голубой**
- **4** .... **красный**
- **5** .... **фиолетовый**
- **6** .... **коричневый/желтый**
- **7** .... **белый**

Добавив 8 к этим цифрам, можно получить «яркую» версию, но отображение ярких цветов зависит от оборудования и на некоторых системах способно привести к миганию. **ПХ**

4 Ох, этот iWireless

Пытаюсь установить OpenSUSE 10.2 на iMac моей внучки. Установка шла хорошо до тех пор, пока я не попробовал добавить беспроводной адаптер PluscomWU-ZD1211B USB: не установился его драйвер, ни из комплекта, ни из сети. При запуске make от имени суперпользователя я получаю множество сообщений: *make [6] Error1, make [5] Error 2, leaving dir* и т.д. В *YaST* адаптер определен как ***USB2.0 Wireless / drivers: active / modules: yes / modprobe: zd1211rw?*** Я скопировал прошивки, как написано в readme, и настроил *ssid* по подсказкам *YaST*. В SUSE 10.2 нет исходников драйверов *zd1211rw* или *ndiswrapper*, и я не могу установить их через *YaST*! Я новичок в борьбе с командной строкой, и надеюсь, вы мне поможете, не то я с ума сойду!

Крис Брайан [Chris Bryan]

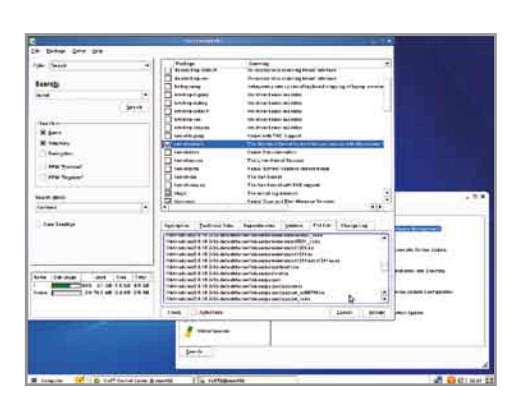
Могу подтвердить, что драйверы *zd1211* работают с ядром PPC, я использовал их с моим iBook до появления драйверов для встроенной беспроводной карты. Но вам не нужно устанавливать какие-либо драйверы для адаптера на базе *zd1211*: они включены в стандартное ядро SUSE 10.2. Запуск следующих двух команд должен подтвердить присутствие модуля *zd1211rw*.

```
modprobe -l | grep zd1211
modinfo zd1211rw
```

Вы также можете просмотреть файлы в *YaST*. Перейдите в раздел Software Management (Управление программным обеспечением), найдите пакет *kernel-default* и нажмите на вкладку Файлы. Спуститесь вниз и найдите соответствующие модули в **drivers/net/wireless**. *Modprobe* запустился без ошибок? Если нет, Вам, возможно, придется позаботиться, чтобы прошивка находилась в нужном месте. Большинство драйверов ожидают найти ее прямо в **/lib/firmware**, но данный конкретный драйвер думает, что она в **/lib/firmware/zd1211**. Самую новую прошивку можно загрузить с **http://sourceforge.net/projects/zd1211**.

Если модуль загружается, а сетевое устройство присутствует в выводе команды

```
ifconfig -a
```



› Как показывает *YaST*, поддержка беспроводных карт ZD1211 уже вскомпилирована в ядро SUSE 10.2.

Коротко про...

DVD-CSS

Смотрите кино из-под Linux

Это из-за системы DVD Content Scrambling System (CSS) большинство коммерческих DVD не проигрывается мультимедийными проигрывателями Linux при установке «из коробки». CSS пресекает копирование DVD-видео за счет шифрования данных на диске. Ключ шифрования хранится в специальной области диска, не доступной стандартным устройствам записи DVD. В пустых DVD +/-RW эта область заполнена нулями. Если сделать прямую копию DVD, зашифрованного CSS, скопируются все зашифрованные данные, но не ключ к шифру.

Можно взломать шифр, используя обычный подбор, поскольку методика шифрования слабая, но на это все же уйдет около суток – слишком долгое время для того, чтобы посмотреть фильм. Но алгоритм CSS был раскрыт тремя людьми. Двое из них предпочли остаться анонимами, третьим был подросток Йон-Лех Йохансен [Jon Lech Johansen], известный как «Йон-DVD». Эти трое создали *DeCSS*, впоследствии включенную в библиотеку *libdvdcss*. Если она установлена, мультимедийные проигрыватели Linux могут использовать ее для воспроизведения зашифрованных DVD.



› *Xine*, да и все другие DVD-плейеры Linux, будут показывать зашифрованные DVD после установки libdvdcss.

Правовой статус DeCSS темен: Йон-DVD даже был однажды арестован, но не осужден, и за использование *libdvdcss* пока никого не посадили. Но риск велик, так что практически нет дистрибутивов, включающих поддержку CSS по умолчанию. Установка библиотеки – задача тривиальная, и она оставлена на усмотрение пользователя. Многие дистрибутивы имеют неофициальные сайты, где есть подходящие пакеты, или же можно установить ее из исходных текстов. Поэтому вы не найдете *libdvdcss* на **LXFDVD**. Установка *libdvdcss* позволяет всем DVD-проигрывателям показывать коммерческие DVD без каких-либо дальнейших действий.

то проблема, скорее всего, в настройке. Прежде всего, отключите шифрование. Конечно, не следует пользоваться незащищенным соединением, но при настройке шифрование только мешает, а пока вы не установили связь, и защищать нечего. Используя *YaST* для конфигурирования сети, попробуйте сделать это и с *Network Manager*, и без него. Одни установки работают через *Network Manager*, а другим лучше живется при настройке стандартным методом. Кстати, если вы в будущем столкнетесь с аналогичными проблемами при сборке из исходных текстов, имейте в виду: номер ошибки ни о чем не говорит, причину ошибки указывает сопроводительный текст.

5 Фото-групповуха

В Мне бы хотелось иметь простую команду для перевода всех RAW-изображений в папке, скажем, **/home/andy/photographs/new**, в формат TIFF со сжатием LZW. Не знаю, важно ли это, но расширение RAW-файлов – PEF (в формате RAW Pentax).

Энди Инглэнд [Andy England]

О Программа для конверсии «сырых» (raw) файлов с цифровых камер называется *dcraw* (www.cybercom.net/~dcoffin/dcraw). Она преобразует RAW-форматы большинства цифровых камер в формат NetPBM PPM (Portable PixMap). Затем понадобится еще одна программа, для преобразования PPM-данных в TIFF. Такое умеют команда *convert* из *ImageMagick* и программа *pnmtotiff* из пакета NetPBM. На вопрос о том, которую предпочесть, проще всего ответить «Ту, что уже установлена». *Dcraw* должна быть в репозитории вашего дистрибутива; в противном случае загрузите исходный код и скомпилируйте/установите ее обычным порядком.

Чтобы не забивать при преобразовании жесткий диск PPM-файлами (PPM –занимающий много места несжатый формат, где используются три байта на пиксель), можно перекинуть вывод *dcraw* прямо в программу конверсии. Для групповой обработки с помощью *ppm2tiff* всей директории с файлами, запустите в терминале

```
for f in ~/photographs/new/*.PEF
do
 dcraw -c "$f" | pnmtotiff -lzw >"${f/.PEF/.tif}"
done
```

Аргумент **-c** велит *dcraw* посылать данные на стандартный вывод, который передается на вход *pnmtotiff*. Выражение **${f/.PEF/.tiff}** подставляет имя исходного файла, меняя расширение **PEF** на **tif**. Чтобы использовать *ImageMagick*, измените третью строку на

```
dcraw -c "$f" | convert -compress lzw – "${f/.PEF/.tif}"
```

В этом случае *convert* использует обычную нотацию **"-"** для стандартного ввода. На man-страницах *imagemagick*, *dcraw*, *netpbm* и *convert* подробно описаны всякие дополнительные опции для тонкой настройки процесса.

Эти команды предполагают нахождение всех PEF файлов в одном каталоге. Если Ваша камера сохраняет их во вложенных каталогах, используйте команду **find**, чтобы создать список имен. Конвертированные изображения можно записать в другой каталог:

```
find /mount/pointof/camera -name '*.PEF' | while
 read f
 do
  dcraw -c "$f" | convert -compress lzw - "~/
 photographs/tiff/$(basename "$f" .PEF).tif"
done
```

Здесь мы используем команду **basename** для извлечения имени файла из полного пути и удаляем расширения. Кавычки вокруг имен файлов нужны на случай, если имя файла или каталога содержит пробелы. **НВ**

6 Обижен GnuCash...

В Я без проблем работал с разными версиями *GnuCash* несколько лет. Однако при использовании версии 2.0.2 с последними релизами Ubuntu и SUSE папка **Current Account** два раза исчезала между записью счета и его рестартом. В первый раз я вернулся к ранее сохраненным счетам и заново набрал мои банковские ведомости. Во второй раз потерял 6/52 данных, и не намерен повторно вводить утраченное. Есть ли способ восстановить данные и/или избежать потерь в будущем? Программа явно непригодна для использования в ее нынешнем виде.

Джо Лэм [Joe Lamb]

О Моя первая реакция была – да я пользуюсь *GnuCash* много лет, сейчас в версии 2.2.1, но такого не видывал; наверняка проблема в конкретных настройках. Но это бы вас не утешило. Точно ли в вашей файловой системе нет ошибок? Запуск *fsck* для раздела **/home** был бы разумным шагом.

Что касается восстановления данных, *GnuCash* сохраняет резервные копии и файлы данных в свой каталог; файлы данных называются **AccountName.datestring.xac**. Найдите самый последний неповрежденный файл и скопируйте его в **AccountName**. Не надо его переименовывать, пусть лучше сохранится резервная копия на случай потери или нового повреждения учетной записи.

Эти резервные файлы создаются при каждом запуске *GnuCash*, и если их там оставлять, заполнят весь ваш жесткий диск, поэтому *GnuCash* удаляет старые файлы по истечении срока, установленного в разделе General окна Preferences. Значение **0** означает, что резервные файлы автоматически удаляться не будут.

Если ваша файловая система в порядке, было бы неплохо разобраться с первичной причиной ошибки »

Часто задаваемые вопросы

Виртуальные частные сети

Можно ли работать из дома, имея доступ к частной сети вашей компании?

» Вроде и круто, но что это?
Виртуальная частная сеть [Virtual Private Network, или VPN] является сетью, туннелируемой через другую сеть. Главное назначение VPN – перенос приватных данных через публичные сети, такие как Интернет. Соединение виртуальной сети проходит через зашифрованный туннель в сети общего доступа. Безопасность последней никак не гарантируется, так как нельзя проконтролировать, кто имеет доступ к промежуточным устройствам; а шифрованные данные бесполезны для всех, кроме вас.

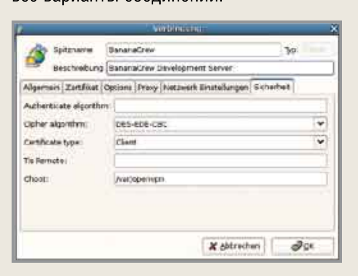
» Разве SSH или HTTPS делают не то же самое?
Эти два типа шифрованного соединения используются для определенной цели: запуска оболочки или защищенного доступа к web-страницам. VPN работает на более низком уровне, шифруя все передаваемое.

» А зачем?
Подсказка – в названии: для поддержания частной сети через общедоступную. Например, у вас в офисе внутренняя сеть, заблокированная брандмауэром, чтобы секреты вашей фирмы не утекли в Интернет. Но необходим доступ к вашей рабочей системе из дома или с дороги. А VPN устанавливает связь между вашим компьютером и сетью компании, и вы работаете в сети точно так же, как сидя за столом в офисе; перед выходом из реальной сети все данные шифруются и проходят через Интернет в целости и сохранности.

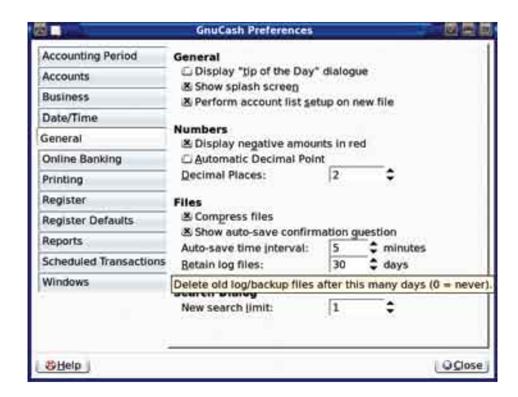
» Нужно ли для этого специальное оборудование?
Потребуются VPN-сервер, запущенный в сети, и подходящая настройка брандмауэра, а также клиент VPN, установленный локально. Это может быть программа на вашем компьютере (единственный возможный вариант в так называемой конфигурации «road warrior», т.е. про работе со своего ноутбука из любой точки мира) или выделенный VPN-маршрутизатор, установленный, например, в удаленном офисе фирмы.

» Наверное, он недешев?
Почему же? VPN поддерживается многими маршрутизаторами, некоторые из них ненамного дороже самых элементарных, без VPN. Остерегайтесь маршрутизаторов, заявляющих поддержку VPN в режиме pass-through: они всего лишь обеспечивают использование VPN-соединения, но не способны управлять им, и вам все равно потребуется держать клиентское ПО VPN на своем компьютере.

» Что это за ПО?
Вам нужны клиентское ПО и ядро соответствующей конфигурации. В большинстве дистрибутивов такое ядро имеется по умолчанию (поскольку большинство дистрибутивов засовывают в свои ядра по умолчанию все, что только может пригодиться). ПО зависит от вида VPN на другом конце соединения, но проекты OpenVPN (http://openvpn.net) и Openswan (www.openswan.org) охватывают почти все варианты соединений.



› Существуют GUI-инструменты для настройки VPN, а можно использовать *Webmin*.



› **Пользователи *GnuCash* должны сохранять резервные копии данных и журналов.**

» (восстановление из резервной копии – это не выход): запустите *GnuCash* из терминала, и при очередном падении Вы увидите некоторые ошибки в выводе терминала или log-файлах. Поищите в списках рассылки и файле bugzilla для *GnuCash* (через www.GnuCash.org) аналогичные проблемы и создайте сообщение об ошибке, если не обнаружите ничего подобного: если разработчики не узнают о проблеме, они ее и не устранят. Было бы также мудро сперва попробовать новую версию: возьмите RPM *GnuCash 2.2.x* для SUSE с http://rpm.pbone.net.

7 Konqueror против USB

Похоже, все дистрибутивы обрабатывают скрипты с USB-брелков по-разному. Попробуем запустить скрипт командой execute в *Konqueror*: Ubuntu отработает без проблем (правда, сперва моя обновленная Feisty по ошибке монтировала диск в режиме non-exec). SUSE 10.2 и Startcom сочтут брелок удаленным и не будут ничего с него выполнять. Fedora 7 Live выдаст запрос на подтверждение, но примется искать скрипт в неправильном каталоге, если *Konqueror* использовался для открытия USB-устройства через значок. Автомонтировщик монтирует **/dev/sdb1** в **/media/disk**, а *Konqueror* видит URL как **/media/sdb1**. При попытке запуска исполняемого файла *Konqueror* пытается перейти в **/media/sdb1** и не может найти исполняемый файл. После перехода в **/media/disk** скрипты выполняются правильно.

Хотя SUSE 10.2 ничего не запускает через *Konqueror*, если на брелке имеется скрипт **autostart.sh**, KDE предлагает выполнить его, стоит лишь вставить брелок. Надо думать, что все это настраиваемо (не вкомпилировано же жестко!), но как? HAL, *udev* и KDE – все причастны, файлов конфигурации уйма, так с которого начать?

Ян Годдард [Ian Goddard]

Причин такого разнобоя несколько: отчасти это из-за особенностей работы KDE, отчасти из-за отсутствия реальных прав доступа на файловой системе FAT. Вы уже обнаружили, что KDE использует для доступа к файлу на съемном устройстве не реальный путь. *Konqueror* отображает **media:/media/имя_устройства**, но фактически монтирует устройство в **/media/метка_тома** (или **/media/disk**, если у диска нет метки тома). Это не вызывает проблем у большинства приложений KDE, так как они понимают URL'ы **media:/** и **system:/media**, зато их не понимают не-KDE программы вроде *bash*. Ubuntu справляется с проблемой, потому что там установлена заплатка, направляющая *Konqueror* прямо в точку монтирования, но пока не решено, не перевешивают ли недостатки данного метода его преимущества.

Существует простой способ решить эту проблему для всех дистрибутивов. Откройте окно свойств, нажав правой кнопкой мыши на скрипте. Нажмите значок справа от надписи Type: Shell Script для открытия редактора типов файлов и, используя кнопку Добавить, внесите *bash* в список программ. Не ищите *bash* в списке программ, введите путь к нему в поле над списком. Подтвердите изменения; теперь при щелчке правой кнопкой мыши на скрипте *bash* появится в подпункте меню Открыть с помощью. Если поставить *bash* в верх списка, вы также сможете запускать скрипты, просто нажав значок: это удобнее, чем выбирать значок, а затем переходить на верхнюю часть окна для поиска меню. Автозапуск **autostart.sh** настраивается в разделе Peripherals/Storage Media Центра управления KDE.

Другая возможность, на которую вы уже указали – редактирование конфигурации HAL, чтобы вместо имени тома при монтировании устанавливалось имя устройства. Правда, это сделает систему менее понятной, особенно при использовании файловой системы FAT, созданной другими устройствами, которая дает им значимые имена: карта памяти моего фотоаппарата всегда идентифицируется по имени, как и мой аудиоплейер, поэтому нет опасности скопировать файлы на неправильное устройство, что легко может случиться при монтировании устройств в **/media/sde1** и **/media/sdf1**. **НБ**

8 Конфуз с установкой

Мой ноутбук имеет двойную загрузку с PCLinuxOS. При каждой попытке установить приложения с **DVDLXF**, следуя инструкциям в журнале, меня озадачивает одна штука: это отсутствие дефиса, показанного в команде распаковки tar-архива. В журнале показано **tar xzvf /mnt**, а на самом деле надо **tar -xzvf /mnt**. Установка ПО в Linux – абсолютная абракадабра, и любые объяснения или оправдания этой ситуации бессмысленны. Лучше бы публиковались конкретные инструкции по установке каждой программы с прилагаемого диска: было бы интересно сравнить установки для 30 различных программ в одном документе. Уверен, их будет 30 разных. Например, я сейчас пытаюсь установить DEFCON с LXF96. Инструкция велит набирать **./configure**, но мой терминал возвращает на это следующее:

```
bash: ./configure: No such file or directory
```

Камерон Райт [Cameron Wright]



Вопрос-победитель (русская версия)

Виктор Басынин получает подарочный сертификат на 1000 рублей от интернет-магазина **LinuxCenter.Ru**! Просим победителя выйти на связь с редакцией: info@linuxformat.ru

★ Проблемы с обновлениями

После обновления ASPLinux 11 до 11.2 (с диска LXF89) менеджер пакетов перестал работать! Он запускается, но потом появляется окошко, в котором сказано, что он не может подключится к сети – и все. Зачем ему сеть? Наверное, где-то в настройках сказано, что пакеты надо искать в Интернете. Подскажите, где искать эти настройки? Ладно, установил Fedora Core 6 с DVD LXF87/88. С менеджером пакетов – та же проблема!

Виктор Басынин

Обновлениями в Fedora Core и, как следствие, в ASP Linux занимается одна и та же утилита – *Yum*, и ее основные настройки хранятся в файле **/etc/yum.conf** и каталоге **/etc/yum.repos.d**. По умолчанию все системы, действительно, настроены на получение обновлений из интернета. Репозитории, в которых будет производиться их поиск, обычно перечислены в отдельных файлах с расширением **.repo** в **/etc/yum.repos.d**. Откройте (от имени root) такой файл в текстовом редакторе. Он имеет формат «Windows .INI», т.е. состоит из секций, начинающихся с названия в квадратных скобках (каждая секция соответствует одному репозиторию), и пар «ключ-значение». Нас интересует ключ **baseurl** – измените его значение на **file://место/куда/монтируется/cdrom**, например **file:///media/cdrom**. Старую директиву закомментируйте знаком **#**, не помешает проделать это и для директивы **mirrorlist**. Обратите также внимание на секции с суффиксом **updates**. Если оставить их без изменения, *Yum* будет пытаться искать обновления в сети; если же проделать описанную выше манипуляцию – все данные будут читаться с диска, но об оперативном устранении уязвимостей в этом случае придется побеспокоиться самому. Имейте в виду, что *Yum* не занимается управлением съемными носителями – вы должны самостоятельно вставить диск с пакетами и смонтировать его, прежде чем откроете Установку/удаление программ. **ВС**





Вопрос-победитель (английская версия)

Неконтактная VMware

Я использую *VMware Workstation* для запуска различных виртуальных машин, и с Windows, и с Linux – для разработки web-сайтов, а также чтобы пробовать разные дистрибутивы. Недавно я утратил способность соединять через SSH хост- (Ubuntu) и гостевую ОС. Команды ssh просто висят, пока я не нажму Ctrl+C.

Я могу соединиться по SSH с хостом или гостевой системой с моего ноутбука в той же сети, могу и подключиться к ноутбуку с хост- или любой из виртуальных машин; похоже, не работает только соединение между хост- и гостевой системой на одной машине. Я пропинговал гостевые системы из хоста и наоборот, и, по-видимому, не работает только SSH. Но я не изменял ничего в моих настройках SSH на хосте, а так как это влияет на все гостевые системы, я не вижу проблемы в настройках. Я использую мост – я всегда так делал, и все работало.

Дэйв Бэрч [Dave Burch]

Я столкнулся с точно такой же проблемой. Сперва я заподозрил SSH, но затем обнаружил, что другие протоколы тоже сбоят. И выяснилось: поломалось все, что использует TCP, причем в связи с обновлением ядра. Изменилась поддержка движка TCP Offload на некоторых сетевых картах (в моем случае, в контроллере Attansic L1 Gigabit). Движок offload передает часть нагрузки по обработке TCP от ядра к контроллеру сетевой карты, что вызывает проблемы, когда сетевой трафик не идет через сетевой контроллер, хотя он предназначен для той же сети – ситуация, возможная только в виртуальных машинах. Мост дает виртуальной машине IP-адрес в той же подсети, что и Ваша локальная сеть, но трафик не проходит через сетевую карту.

Решение простое: отключите некоторые функции движка offload с помощью *ethtool*. Все потенциальные источники проблем одновременно ликвидируются так:

```
sudo ethtool --offload eth0 rx off tx off sg off tso off
```

Если это работает, неповинных Вы можете «помиловать». На моей системе я отключил только самое необходимое:

```
sudo ethtool --offload eth0 tx off
```

Определив, что именно надо отключать, можно добавить команду в **/etc/rc.local** (без части sudo), и она будет выполняться автоматически при загрузке. **НБ**



Установка ПО в обход менеджера пакетов вашего дистрибутива может вызвать путаницу, потому что есть различные способы поставки ПО. Общие инструкции в журнале распространяются на некоторый процент ПО – а именно, на программы, поставляемые в виде исходных кодов с использованием набора *autotools*. Во многих случаях имеются инструкции на диске, в виде **readme**- или **install**-файлов, сопровождающих ПО, и они часто содержат идентичные инструкции. Их следует рассматривать как «Евангелие от Программиста» по установке.

Дефис перед аргументом был нужен только в ОЧЕНЬ старых версиях *tar*. По сути, опция z давно уже лишняя, так как tar распознает распространенные форматы сжатия. Я обычно использую tar xf filename.

. (точка) обозначает текущую директорию: например, ./configure сообщает оболочке о необходимости запуска программы или скрипта *configure* в текущей директории. Как только вы это поймете, понятна станет и причина ошибки: нет скрипта конфигурации этой программы. Дело в том, что DEFCON не поставляется в виде исходных текстов: это двоичный пакет, не требующий установки. Вместо скрипта *configure* вы увидите скрипт *defcon*; для его запуска просто наберите ./defcon.

Я согласен, что это не совсем очевидно, приложенная документация подразумевает использование программы под Mac OS, но это недочет одной конкретной программы, хотя мы могли бы, пожалуй, отметить это в журнале. Программы данного типа выигрывают от использования системы *Autopackage* (www.autopackage.org), которую пропагандирует Майк, web-мастер сайта Linux Format и редактор DVD. *Autopackage* пакует программное обеспечение в единый файл, который вы «запускаете» для установки ПО. **НБ**

› **DEFCON поставляется в виде готового к запуску двоичного файла - установка не требуется.**

9 Walkman зашел в тупик

У меня плейер Sony NW-E507 Walkman и Ubuntu 7.04 Для установки и удаления музыкальных файлов и подкастов (что я, в основном, и делаю) этот Walkman использует *Sonic Stage*. Я попытался применить *Mbnet*, но безуспешно. Можно ли использовать этот Walkman с моей ОС?

Глен Кохун [Glen Cohoon]

Устройство Walkman тесно связано с технологией DRM от Sony, так что без *Sonic Stage* не обойтись. Можно монтировать плейер как обычный USB-накопитель и загружать файлы, но они не будут проигрываться. Устройство работает только под Windows, и даже там оно приковано к ПО от Sony; пользователям Mac оно также недоступно. Стандартные MP3-плейеры нынче дешевы, и лучшее, что вы можете сделать – заменить этот упертый Walkman на что-нибудь другое и обращаться с музыкой, которую Вы купили, как вам заблагорассудится. **ПХ**

10 Бридж в одиночку

Вы не знаете, есть ли игра бридж для Linux? Все найденные мною версии предназначаются для игры через Интернет с партнерами.

Смарагда Константиниду [Smaragda Konstantinidou]

Изрядно пошарив, мы отыскали всего одну игру, где можно играть против компьютера, но она не с открытым кодом. Значит ли это, что программисты Open Source не играют в бридж, или что они предпочитают играть с реальными оппонентами? Неплохая тема для явно бесцельного социологического исследования; ну да Вам это ни к чему. Обнаруженная коммерческая игра – *Xcontractbridge*, она поставляется на компакт-диске за $29. Более подробная информация – на www.xcontractbridge.com/xbridge.htm, там есть и демо-версия для скачивания. Это заранее собранный двоичный файл, который должен работать на большинстве i386 + или x86_64 систем и вроде бы «бегает» без проблем на моей машине Core2Duo; правда, я-то в бридже умею только различать красные и черные масти (да и то благодаря наследственности: мой отец был мастером этой игры). **НБ**

11 Двое на одного

Недавно увидел программу *ACTEP* для Microsoft Windows. Она позволяет независимо работать на одном компьютере нескольким пользователям: мониторы подключаются к видеокарте с двумя выходами (или используются две видеокарты), а »

» дополнительные клавиатура и мышь подсоединяются к шине USB. Возможно ли сделать такое в Linux?
Gates Alex

О Возможность создания нескольких рабочих мест на базе одной вычислительной машины существовала в операционных системах Unix изначально. Unix-серверы, обслуживающие один или несколько десятков текстовых терминалов (таких, как *vt100*), когда-то были распространенным явлением. Разумеется, эти возможности не исчезли с распространением X Window System и существуют и в Linux.

Принцип работы работы такой «многоголовой» системы прост. X-сервер (правда, для *X.Org* потребуется дополнительная надстройка в виде *Xephyr*, *XFree* в этом смысле был более продвинут) может обслуживать несколько экранов и получать события от нескольких устройств ввода. Мы можем установить в компьютер несколько видеокарт (конечно, это должны быть PCI-карты, если только у вас нет нескольких AGP-слотов; кроме того, вы можете использовать и встроенное видео), а также подключить необходимое количество клавиатур (устройства **/dev/input/eventX**) и мышей (**/dev/input/mouseX**), используя обычный USB-концентратор. После этого достаточно просто установить соответствие между клавиатурами, мышами и мониторами в секциях ServerLayout файла **xorg.conf** (по одной на каждого пользователя) и запустить несколько копий *X* с различными значениями параметра -layout – именно он указывает, какая секция будет использоваться для данного экземпляра X-сервера. **ВИ**

12 Сумасброд su

В Я установил ALT Linux 4.0 и у меня возникла смешная проблема. Запускаю консоль, набираю *su* и вижу вот что:

```
[andrew@and ~]$ su
password:
[root@and andrew]# echo ~
/home/andrew
[root@and andrew]#
```

Если верить этому, то получается, что root и andrew (я) используем одну и ту же домашнюю директорию?!
Андрей Кондратьев

О Нет, хотя вы с root'ом – одно и то же лицо, домашние директории у вас, скорее всего, разные. «Оптический обман», который вы наблюдаете, происходит из-за того, что по умолчанию *su* меняет лишь идентификатор пользователя, но не его окружение. Таким образом, переменная HOME, в которой хранится путь к домашнему каталогу, остается прежней. Чтобы «имитировать» вход нового пользователя в систему, используйте команду *su -*. **ВС**



Нужна помощь!

› Для наилучшего ответа на ваш вопрос нам нужно знать как можно больше подробностей. Детально опишите конфигурацию системы. Если вы получили сообщение об ошибке, приведите текст сообщения и точно опишите вызвавшие его действия. Если у вас проблемы с оборудованием, то опишите его. Если Linux уже запущен, то выполните в root-терминале следующие команды и прикрепите к письму файл **system.txt**:

```
uname -a >system.txt
lspci >>system.txt
lspci -vv >>system.txt
```

› Пожалуйста, помните, что сотрудники журнала НЕ являются авторами или разработчиками Linux, любых пакетов или дистрибутивов. Зачастую люди, отвечающие за приложения, выкладывают большую часть информации на web-сайты. Попробуйте почитать документацию!

Мы стараемся ответить на все вопросы. Если вы не нашли ответ на свой, это, возможно, потому, что мы уже ответили на похожий вопрос.

Большой вопрос

Как читать почту на любом компьютере в сети, обходясь без MTA.

13 Linux как почтовый прокси

В Я хочу автоматически качать сообщения с моей учетной записи в Интернете и получать их на свой ноутбук с Windows от машины с Linux, чтобы скачивать их только на одну из машин, но чтобы сообщения также остались на моем сервере. Я бы мог просто изменить запись MX, указав на мою машину, но мой провайдер не предоставляет статического IP-адреса (кроме *dyndns*).

Tfg, из форумов LXF

О Запуск местного MTA (Mail Transport Agent) без статического IP-адреса – вещь рискованая: из-за изменения динамического адреса ваша почта может угодить в чужие руки. Некоторые Интернет-провайдеры блокируют порт 25, это потенциальный пункт входа для предполагаемых злодеев. Но для достижения ваших целей свой MTA запускать вовсе не нужно. При помощи *fetchmail* получите почту с вашего почтового сервера, сохраните ее локально, а затем запустите POP3 или IMAP сервер для работы с почтой на локальных ПК. В текстовом редакторе создайте в Вашем домашнем каталоге файл **.fetchmailrc** такого содержания:

```
set daemon 300
poll mail.myisp.com with proto POP3
    user ‘myispuser’ there with password
‘mypass’ is ‘myuser’ here options keep
mda ‘/usr/bin/procmail -d %T’
```

и установите права на чтение только для себя:

```
chmod 600 ~/.fetchmailrc
```

Первая часть велит *fetchmail* работать в режиме демона и опрашивать Ваш почтовый сервер каждые 300 секунд, вторая связывается с почтовым сервером, используя имя пользователя и пароль, и сохраняет почту под местным именем пользователя. Опция **keep** оставляет почту на сервере (убедившись, что все работает, можете ее удалить). Последняя строка предписывает *fetchmail* доставлять почту через *procmail*, а не *sendmail*. Велите *procmail* доставлять почту, поместив в **~/.procmailrc** строки

```
MAILDIR=/var/spool/mail
DEFAULT=$MAILDIR/$LOGNAME/
```

Оконечный **/** обязателен, он приказывает *procmail* использовать для хранения *maildir*, которая позже понадобится IMAP-серверу. Создайте почтовую директорию

```
mkdir -p /var/spool/mail/myuser
chown myuser:mail /var/spool/mail/myuser
chmod 770 /var/spool/mail/myuser
```

Протестируйте все это командой **fetchmail --daemon 0 -v**. Она отключает фоновый режим и показывает все, что происходит. Если все в порядке, установите *fetchmail* на автоматический запуск, выбрав System > Preferences > Personal > Session, нажав New в вкладке Startup Programs и введя **fetchmail** в обоих полях. *Fetchmail* теперь будет запускаться при каждом запуске Gnome. Добившись получения почты в свою систему, обзаведитесь IMAP-сервером для доступа к ней из вашей локальной сети. После установки *Dovecot* из репозиториев Fedora, внесите изменения в **/etc/dovecot.conf**. Найдите строки

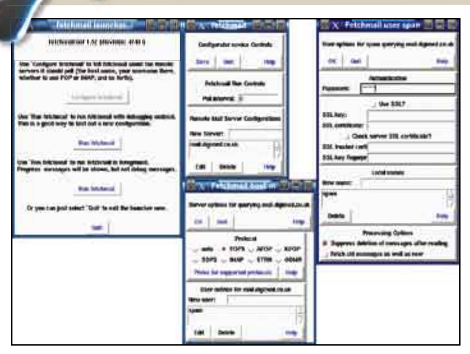
```
#listen= [::]
#mail_location =
```

и замените их на

```
listen = *
mail_location = /var/spool/mail/%u
```

Кроме того, необходимо разрешить порты 110 (POP3) и 143 (IMAP) через брандмауэр. Для исходящей электронной почты, на каждом компьютере можно оставить возможность общаться непосредственно с вашим провайдером через SMTP-сервер. **НБ** LXF

Шаг за шагом: соединяемся с phpMyAdmin



1 Настроим fetchmail

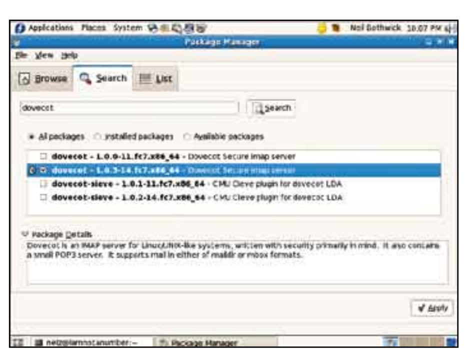
Fedora 7 не включает *fetchmailconf*, и придется настроить *fetchmail* вручную. Другие дистрибутивы предлагают вариант в духе «укажи и щелкни».

2 Протестируем fetchmail

Запуск *fetchmail* не в режиме демона, с подробным выводом, покажет, хорошо ли все работает.

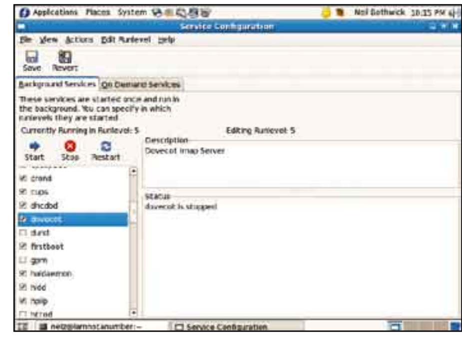
3 Автоматически запустим fetchmail

Через Gnome Session Manager добавьте *fetchmail* в список программ для автоматического запуска при старте рабочего стола.



4 Установим Dovecot

Fedora Core 7 не устанавливает IMAP-сервер по умолчанию, поэтому установите его с помощью Add/Remove Programs.



5 Запустим Dovecot

После пары изменений настроек, используйте Service Configuration для запуска *Dovecot* и добавьте его в список приложений, *запускаемых* при загрузке.



6 Разрешаем доступ

По умолчанию настройки брандмауэра разрешают доступ к почтовому серверу только на локальном компьютере. Откройте порты POP3 и IMAP, чтобы остальные ваши ПК могли им пользоваться.

LXF DVD99

Размечтались об Ubuntu пошустрее? Настроились покодировать? Здесь есть всё...



Майк Сондерс любовно подбирает содержимое диска Linux Format, а также поддерживает сайт www.linuxformat.co.uk.

Патентованная негодность

Мы в **LXF** все обожаем программировать: Ник – дока в Python, Грэм балуется с C++, Пол – наш местный гуру в C#, а я не могу удержаться от первозданного хакерства на ассемблере. Linux – отличная платформа для разработки: практически любой язык имеет свою реализацию в свободном ПО, и документации в интернет – навалом. Золотые времена для кодеров.

Все портит, конечно, проблема патентов на ПО. Большинству программистов кажется абсурдным патентование списка инструкций – метода, как что-то сделать. С чего начать? Что получится в конце? Можно ли запатентовать пузырьковую сортировку – алгоритм, который используется в самых элементарных учебниках программирования? Некоторые законоведы будут спорить, что суммы, затраченные на исследования и разработку программ, оправдывают наличие патентов. Но не так все просто: это трата времени и денег, потому что компании считают более целесообразным приобретать патенты, а не писать программы. Разработчики не могут свободно создавать код из-за боязни ненароком нарушить какой-нибудь патент, а ежели компания-нарушитель угодит под суд, то последует долгий и разорительный процесс.

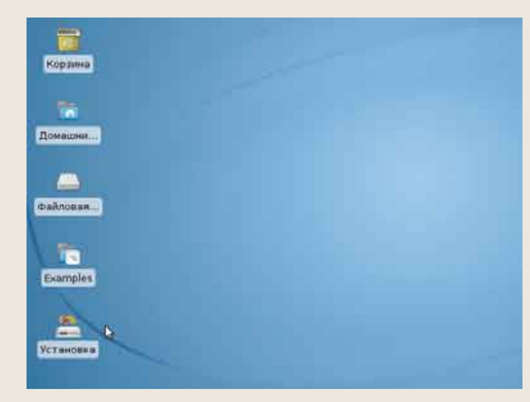
Понимаю, что сейчас я пытаюсь обратить в свою веру единоверцев, но реформа патентного законодательства назрела уже давным-давно. Мелочность некоторых патентов просто поражает: возьмите «1-клик» от Amazon – система онлайн-приобретения товаров одним щелчком мыши. О-о, ну кто бы мог додуматься до столь революционной идеи! Будем надеяться, что грядут перемены, и в будущем мы сможем наслаждаться плодами свободного ПО. А пока что наслаждайтесь сказочными Xubuntu 7.10, SmoothWall Express 3.0 и мега-подборкой языков программирования на DVD!

mike.saunders@futurenet.co.uk

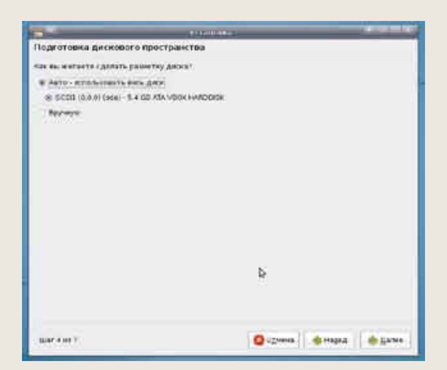
Шаг за шагом: Устанавливаем Xubuntu 7.10



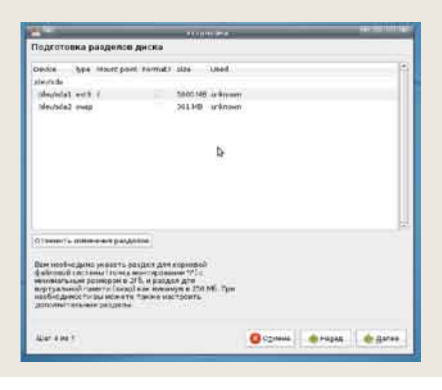
1 Загрузка
Загрузите свой ПК с **LXFDVD** (или с образа на CD-R, если у вас AMD64), и увидите это меню. Нажмите Enter для запуска Xubuntu, или, в случае проблем, выберите безопасный графический режим.



2 Рабочий стол
Перед вами – рабочий стол. Можете познакомиться с дистрибутивом, работающим с DVD. Если хотите установить его на жесткий диск, дважды щелкните мышью по значку Установка в нижней левой части рабочего стола.



5 Разбиение на разделы
Теперь выделите место на диске под Linux (рекомендуем не менее 5 ГБ). Можно выполнить авторазбиение или проделать всю процедуру вручную.



6 Вручную
Если вы создаете разделы вручную, выделите корневой раздел размером не менее 5 ГБ в формате ext3 и раздел подкачки для виртуальной памяти размером 512 МБ.

Дистрибутив Linux

Xubuntu 7.10

Плюсы Ubuntu/Kubuntu: они отлично сконструированы на основе Debian, и за ними стоят превосходные сообщества, обеспечивая им поддержку. **Минусы Ubuntu/Kubuntu:** это не самые быстрые дистрибутивы, особенно на старых машинах. Gnome, KDE и *OpenOffice.org* – известные пожиратели памяти: в эпоху конца 90-х они могли дико тормозить ПК, проводя большую часть времени в разделе подкачки. А еще Gnome и KDE увеличивают время загрузки, если у вас не очень быстрый жесткий диск или процессор. Переключите рабочий стол на более легкую альтернативу Xfce, и что у вас получится? Xubuntu! Он отлично работает на ОЗУ в 128 МБ (K/Ubuntu подавай не менее 256), а значит, это отличный выбор для старых машин. Однако он не ориентирован только на технику вчерашнего дня: если вы фанат Ubuntu и хотите максимально ускорить свой ПК, это опять-таки отличный выбор.

Основанный на Ubuntu 7.10, этот релиз включает некоторые более резвые программы. У вас будут *AbiWord 2.4.6* и *Gnumeric 1.7.11* (*OpenOffice.org* тоже есть, если надо), а для интернета – *Firefox* и *Thunderbird 2.0.0.6*. Да, *Thunderbird* не блещет скоростью, но зато и не зависит от своры библиотек Gnome, как *Evolution* в Ubuntu. Для воспроизведения медиа имеется *gxine*, для редактирования изображений – *GIMP*, а еще есть разнообразные инструменты настройки системы – все на основе ядра 2.6.22. Xfce – это рабочий стол на базе *Gtk*: команда Xubuntu создала Gnome-подобную раскладку с панелью задач внизу и переключением виртуальных десктопов. Файловый менеджер Xfce, *Thunar*, не готов тягаться с *Nautilus* функционально, но зато он куда быстрее и менее ресурсоемок.

Xubuntu 7.10 работает в режиме Live непосредственно с нашего DVD: загрузите его с диска, и вы попадете на рабочий стол. Потом можете поиграть с ним и установить его на жесткий диск, дважды щелкнув по значку Install на рабочем столе (см. пошаговое руководство внизу). Чтобы установить его с Live DVD, вам потребуется 192 МБ ОЗУ; после установки вполне хватит 128 МБ и процессора 500 МГц.

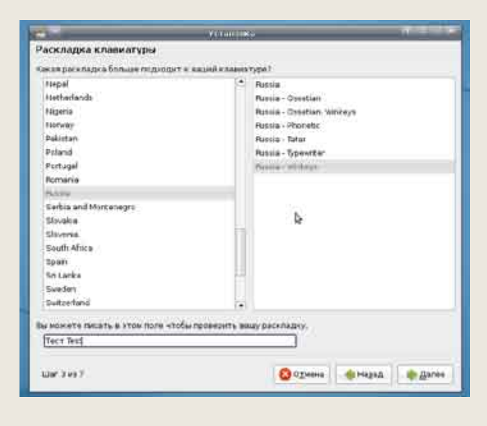
Если вы хотите установить его на машину, где всего 128 МБ ОЗУ, запишите на диск образ **xubuntu-7.10-alternate-i386.iso** из раздела **Дистрибутивы/Xubuntu** нашего DVD. Это позволит установить Xubuntu 7.10 через текстовую систему и потребует всего 64 МБ. А если вы работаете на машине AMD64 и хотите выжать из вашего процессора все до капли, у нас есть 64-битный образ для записи на CD-R и последующей загрузки! Учтите: Xubuntu, как и Ubuntu, не использует пароля root. Если вы хотите открыть root-сессию, введите в терминале *sudo bash*, а затем свой пользовательский пароль. »



› **Мышь-талисман Xfce неплохо вписалась в логотип Xubuntu.**



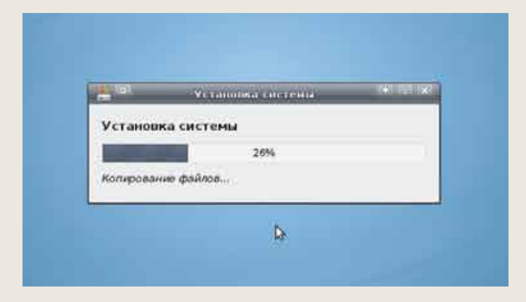
3 Язык
Запустится программа установки, и предложит вам выбрать язык. Затем нажмите Вперед и выберите свой часовой пояс на карте мира.



4 Клавиатура
После этого выберите раскладку клавиатуры из списка: выберите страну слева, а потом – предлагаемые для этой страны варианты раскладки клавиатуры справа.



7 Настройка
Программа установки предложит вам перенести любые настройки Windows, если у вас двойная загрузка, а затем предложит вам создать пользователя и задать пароль. Не забывайте их!



8 Копирование
И, наконец, программа установки начнет копировать файлы Xubuntu на жесткий диск. Покончив с этим, она перезагрузит компьютер – извлеките диск, и наслаждайтесь новой системой Xubuntu!

Дистрибутив-брандмауэр

SmoothWall Express 3.0

Вы не пробовали ввернуть шуруп молотком? Или включить свет поленом? А ведь именно так порой применяют универсальные дистрибутивы Linux, решая совершенно конкретную задачу. И чаще всего это наблюдается в такой сфере, как брандмауэры. Как правило, мы стараемся настроить простой брандмауэр на отдельно взятой машине, чтобы порты при случае не остались открытыми и к вам не просочился бы никакой взломщик. Но если ваш брандмауэр обслуживает несколько машин – например, как переходное звено между вашими системами и огромным злобным Интернетом – нужно уже нечто посолиднее.

Здесь идеально подходят старые машины, непригодные для решения современных настольных задач. Беда в том, что основные дистрибутивы, всякие SUSE, Ubuntu, Fedora и пр. – явный перебор: зачем вам тысячи пакетов или **init**-скрипты для загрузки модулей 3D-видеокарт, если нужен просто брандмауэр? Конечно, можно обкорнать под эту задачу дистрибутивы вроде Debian и Slackware, но гораздо лучше бы сразу запустить специальный, заранее настроенный дистрибутив. Так вот вам SmoothWall Express 3.0: компактный дистрибутив Linux, работающий как межсетевой экран. А главное, он может обходиться без монитора, поскольку вся пост-установочная настройка производится через браузер на другой машине. SmoothWall настраивает простой web-сервер для интерфейса настройки, и всякий раз, когда вы в сети, вы можете администрировать свой брандмауэр через браузер.

Системные требования – Pentium 200 МГц или выше, 128 МБ ОЗУ и 2 ГБ места на жестком диске. Для установки понадобятся экран и клавиатура – но потом можно их убрать, если вашему брандмауэру предстоит участь ящика в углу. Чтобы установить дистрибутив, запишите образ CD **smoothwall-express-3.0-i386.iso** из раздела **Дистрибутивы** нашего диска на CD-R, затем загрузите компьютер с него. (Вам надо записать его как ISO-образ, а не просто скопировать на болванку.)

SmoothWall использует простой текстовый установщик – если вам случалось устанавливать старые версии Debian или Slackware, он вас не смутит. Кто знаком только с программами установки с графическим интерфейсом – следуйте подсказкам, и все будет хорошо. Помните: SmoothWall не предусматривает инструментов разбиения диска на разделы – он запишется поверх всех данных на вашем жестком диске. И если на жестком диске хранится что-то важное, сперва сделайте копию!

Чтобы начать процесс, следуйте приведенной ниже инструкции. Запишите и загрузите образ CD, который далее скопирует SmoothWall на жесткий диск. Затем он предложит вам настроить сетевое соединение – запомните IP-адрес, который вы дадите своей машине, если будете настраивать ее вручную. Проделав все это, в ответ на соответствующее предложение перезагрузите компьютер; после чего можете покинуть свою машину-брандмауэр, а на другом компьютере в сети открыть в браузере http://smoothwall:81 или https://smoothwall:441 (если не сработает, введите http://<IP-адрес>:81, используя IP-адрес, установленный во время настройки). Теперь можно настраивать брандмауэр через браузер как угодно!

Знакомимся с интерфейсом SmoothWall

1. Разделы
Можете переключаться между текущим статусом, системными настройками и проверкой журналов.

2. Интерфейсы
Эта панель показывает ваш локальный и удаленный IP-адреса и отображает статистику за день/месяц.

3. Статус
В этой строке показано время работы машины и нагрузка на нее.

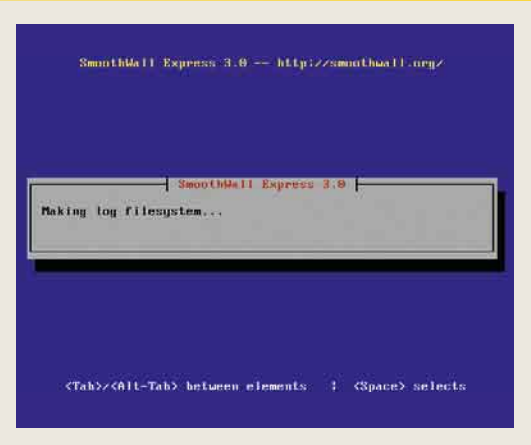
Выключение
Если вам в силу каких-то причин надо перезагрузить брандмауэр, просто нажмите здесь.

График
Этот график наглядно демонстрирует кривую сетевого трафика, помогая отслеживать пики активности.

Шаг за шагом: Устанавливаем SmoothWall Express 3.0



1 Загрузка
Запишите ISO-образ CD из нашего раздела **Дистрибутивы** на CD-R, затем загрузите с CD ваш компьютер. Читайте информацию на экране загрузки, и затем нажмите Enter. Полезные форумы вы найдете на http://community.smoothwall.org.



2 Установка
Следуйте подсказкам, и SmoothWall скопируется на ваш жесткий диск. Помните, что в нем нет встроенной опции разбиения диска на разделы, поэтому он запишет все данные прямо на жесткий диск! Если у вас там важные и нужные файлы, заранее сделайте копии.

Офисный пакет

OpenOffice.org 2.3

OpenOffice.org теперь выходит в виде и RPM-, и **.deb**-пакетов [это не новость для пользователей сборки «Инфра-ресурса», но настала пора-пора-порадоваться и за наших зарубежных коллег, – *прим. ред.*]. Раньше в системах на основе Debian или Ubuntu приходилось конвертировать RPM пакеты в **.deb** с помощью утилиты командной строки alien. А теперь вы просто выбираете соответствующий архив для вашего дистрибутива, распаковываете его и устанавливаете пакеты. А также:

» **Writer** теперь экспортирует в формат MediaWiki (система разметки, используемая в Wikipedia), так что если вы считали его слишком трудным или странным, просто печатайте ваш текст в *OOo* и экспортируйте его.

» **Calc** поддерживает константы-матрицы/массивы в формулах, имеет горячие клавиши для вставки/удаления ячеек (Ctrl+= и Ctrl+-), и выводит более информативные сообщения об ошибках.

» **Charts** теперь может создавать гистограммы и круговые диаграммы за меньшее количество шагов, чем раньше. Палитра по умолчанию обогатилась новыми цветами, трехмерные схемы выглядят лучше, появился новый вид разделенной круговой диаграммы.

» **Base:** написан новый генератор отчетов (в качестве расширения), заодно прибавилось горячих клавиш.

» **Impress** научился использовать графику PNG при экспорте в HTML. Он также позволяет воспроизводить аудио во время слайд-шоу, и вы можете редактировать путь анимации для движущихся объектов на слайдах.

Рассмотрим, как все это запустить из раздела **Офис** нашего DVD. Прежде всего, с помощью менеджера пакетов вашего дистрибутива удалите имеющуюся у вас версию OOo, во избежание конфликта новой и старой версий. Очистив систему, скопируйте **OOo_2.3.1_LinuxIntel_install_ru.tar.gz** для дистрибутивов на основе RPM (а именно Fedora, Mandriva, SUSE) или **OOo_2.3.1_LinuxIntel_install_ru_deb.tar.gz** для Debian и Ubuntu в домашнюю директорию. Для RPM-пакетов откройте терминал и введите:

```
tar xfvz OOo_2.3.1_LinuxIntel_install_ru.tar.gz
cd OOG680_m9_native_packed-1_ru.9238/RPMS
su
rpm -ivh *.rpm
```

Архив **.tar.gz** распакуется, и произойдет установка содержащихся в нем RPM-пакетов. Учтите, что после ввода третьей команды, переключающей на пользователя root, потребуется ввести пароль. Теперь каталог **OOG680_m9_native_packed-1_ru.9238** и файл **OOo_2.3.1_LinuxIntel_install_ru.tar.gz** можно удалить из вашей домашней директории. Для запуска *OOo 2.3*, введите:

```
/opt/openoffice.org2.3/program/soffice
```

Можете завести ссылку на эту программу с рабочего стола или из меню. В системах на основе Ubuntu и Debian, введите:

```
tar xfvz OOo_2.3.1_LinuxIntel_install_ru_deb.tar.gz
cd OOG680_m9_native_packed-1_ru.9238/DEBS
```

Затем, в Debian, введите *su*; в Ubuntu введите *sudo bash*. Вам предложат ввести пароль. И, наконец, введите:

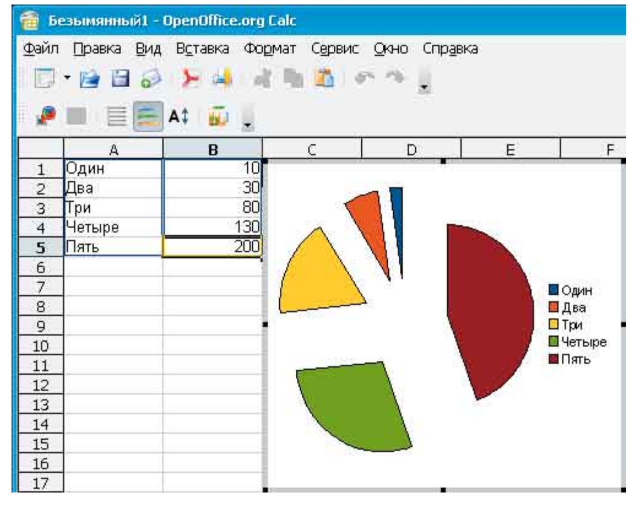
```
dpkg -i *.deb
```

для установки **.deb**-пакетов. Теперь можно запустить программу, введя

```
/opt/openoffice.org2.3/program/soffice
```



› Все еще тормозит? Отключите Java через меню Сервис > Параметры, и скорости прибавится.



› Компонент создания диаграмм переделан, и теперь, кроме всего прочего, можно создавать разделенные круговые диаграммы.



3 Настройка
После установки вы можете настроить сеть, порты по умолчанию и пароль пользователя root. Затем система перезагрузится, и SmoothWall запустится с жесткого диска. С другой машины зайдите на http://smoothwall:81, чтобы настроить дистрибутив.

Другие программы

Новые релизы

Предлагаем вам обратить внимание на *Willow* – оригинальный прокси-сервер с фильтрацией контента, который можно найти в разделе Интернет. Вместо традиционных белых и черных списков, а также регулярных выражений, *Willow* использует алгоритм Байеса, успешно применяемый почтовыми клиентами для борьбы со спамом. *Willow* уже умеет отличать «хорошее» от «плохого» в соответствии с предпочтениями Woodland Hills School District (Питсбург, штат Пенсильвания), но вы легко сможете скорректировать его поведение или произвести обучение с нуля. Сервер написан на Pyhton, поддерживает кэширование и управляется через web – чем не находка для школы?

Кроме этого, есть еще множество чудесных новинок из мира открытого кода. Вышла новая версия Linux From Scratch (6.3): это и справочник по созданию дистрибутива, и набор программ, позволяющий самому создать систему Linux с нуля. Кто давно уж хотел разобраться, как работает система Linux, и мечтает о новом проекте, скорее попробуйте! В разделе **Дистрибутивы** вы найдете полное руководство в форматах HTML и PDF, плюс исходный код и все необходимые заплатки.



› Flash: классный просмотр игровой классики на YouTube!

Достойны рассмотрения и новые релизы *AbiWord* и *Gnumeric* в разделе **Рабочий стол**, и *Flash Player 9* в разделе **Интернет**. Знаем-знаем, не все любят сайты, обвешанные Flash, но охотникам побродить по YouTube он пригодится. В разделе Справка вам предлагается RUTE, руководство по администрированию системы, плюс мега-подборка советов экспертов в разделе **Ответы**: целый воз проблем читателей **LXF** по поводу Linux, и все они решены!

ПО для программистов

Мегапак для разработки

Мы добыли самый свежий релиз GCC, 4.2.2. Этот массивный пакет компиляторов включает C, C++, ADA, FORTRAN и другие языки, и является фактическим стандартом для создания программ для Linux с помощью C/C++. В репозиториях пакетов вашего дистрибутива GCC почти наверняка есть, но если вам захотелось чего-то новенького, можете взять исходный код с нашего диска.

Для поклонников других языков есть Python 2.5.1, Perl 5.8.8 и Ruby 1.8.6. Кто следил за серией руководств Пола по C# в прошлых номерах **LXF** и понял, что его версия Mono не самая новая – берите Mono 1.2.5.1 и MonoDevelop 0.17 (1.0 Beta2). Последний поставлен в виде исходного кода, а первый – в виде опрятных двоичных пакетов: запустите их, и они установятся сами. Скопируйте **mono-1.2.5.1_3-installer.bin** в домашнюю директорию, затем откройте терминал, переключитесь на root и введите:

```
chmod +x  mono-1.2.5.1_3-installer.bin
mono-1.2.5.1_3-installer.bin
```

После установки вы найдете новые файлы Mono в директории **/opt**. Сюда включены .NET-совместимая среда выполнения, *mcs* (компилятор C#) и привязки для графических программ *Gtk#*. Возможно, вы захотите добавить путь к двоичным файлам Mono в **/opt** в переменную среды $PATH, чтобы не печатать каждый раз длинные команды.

Lingua esoterica

Многим из вас, похоже, пришлись по душе наши публикации про экзотические языки программирования (LXF87 – LXF90), ну так вот вам еще. При нашем дефиците места мы, понятно, не в состоянии привести полное руководство по работе с ними, но покажем, как они выглядят и в каком направлении искать, если вы захотите узнать больше. Первый: OCaml (Object Categorical Abstract Machine Language) – это функциональный язык программирования со статической типизацией и автоматическим управлением памятью. В основном используется в академических кругах; его главная забота – производительность. Пример Hello world:

```
print_endline "Hello world!";;
```

Руководство имеется на www.ocamltutorial.org. Еще у нас есть Eiffel, объектно-ориентированный язык, легко читаемый, с акцентом на повторное использование кода:

```
class
    HELLO_WORLD
create
    make
feature
    make
        do
            io.put_string ("Hello, world!")
            io.put_new_line
        end
end
```

Краткий обзор вы найдете на http://tinyurl.com/2rp6w2. Если вас интересует искусственный интеллект, познакомьтесь с GNU Prolog, свободной реализацией языка Prolog. Это логический язык: он не дает компьютеру пошаговых инструкций для выполнения, вместо этого вы заваливаете компьютер данными и используете его для обработки. Например:

```
cat(tom) :- true.
?- cat(tom).
```

Первая строка сообщает, что имеется кот по имени Том, а вторая строка спрашивает: «Есть ли кот по имени Том?» Интерпретатор отвечает: «Да». Судя по всему, его вряд ли можно назвать идеальным для создания web-серверов или игр, но когда вы имеете дело с большими объемами информации, например, база знаний, он очень полезен. На http://tinyurl.com/2kdmy4 найдется подробное и полезное руководство.

И, наконец, взгляните на Gforth – версию FORTH, языка программирования старой школы (он появился в 1960-е), который по-прежнему уважают в научной среде. Обратите внимание на GCL – это GNU Common Lisp (имейте в виду: программы Lisp сначала могут казаться абсолютно нечитаемыми), и OpenCOBOL для тех, кто находит удовольствие в классических бизнес-мейнфреймах и IDENTIFICATION DIVISION. А если вас гнетет тоска по старым добрым временам Speccy и C64, вашу ностальгию излечит FreeBASIC!

```
(defun queue-next (queue ptr)
  "Increment a queue pointer by 1 and wrap around if needed."
  (let ((length (length (queue-elements queue)))
        (try (the fixnum (1+ ptr))))
    (if (= try length) 0 try)))

(defun queue-get (queue &optional (default nil))
  ; return DEFAULT if the queue is empty."
  "Get an element from QUEUE
  (check-type queue queue)
  (let ((get (queue-get-ptr queue))
        (put (queue-put-ptr queue)))
    (if (= get put)
        ;; Queue is empty.
        default
        ;; Get the element and update the get-ptr.
        (prog1
          (svref (queue-elements queue) get)
          (setf (queue-get-ptr queue) (queue-next queue get))))))
```

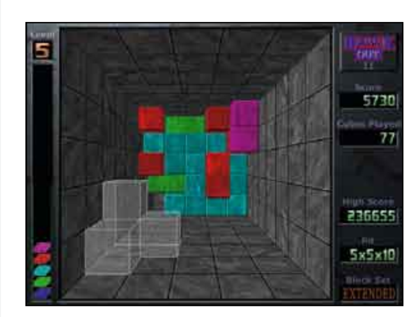
› Вот пример стандартного кода Lisp. Дрожите!



› **Помните *AMOS* и *Blitz*? На FreeBASIC можно писать собственные игры вроде этой, *Space Impakto*.**

И напоследок...

Если создание собственных игр для вас все же экстрим, не грустите: мы тщательно отобрали самые лучшие и свежие релизы, способные вас развлечь. Интригующий вариант классического *Тетриса* – *Blockout II*: все те же классические падающие блоки, но только в трехмерном варианте. Поклонники аркадных стрелялок должны попробовать *Powermanga*, о которой мы рассказывали в разделе **HotPicks** выпуска LXF81. Если вам по-настоящему нравится сверхпопулярная стратегия *Colonization* Сида Майера [Sid Meier], взгляните на *FreeCol* – это ее клон с открытым кодом, написанный на Java, и очень симпатичный. И напоследок – при желании сразиться с приятелями в сети, не пропустите *netPanzer*, войнушку для нескольких игроков со скоростными баталиями и игровом процессе в стиле нон-стоп. Ба-бах! LXF



› **Для победы в *Blockout II* надо заполнить все слои, а не просто линии, как в обычном *Тетрисе*.**

ВНИМАТЕЛЬНО ПРОЧТИТЕ ЭТО ПЕРЕД ТЕМ, КАК ИСПОЛЬЗОВАТЬ DVD-ДИСК !

Информация о диске

ЧТО-ТО ПОТЕРЯЛИ?

Часто случается, что новые программы зависят от других программных продуктов, которые могут не входить в текущую версию вашего дистрибутива Linux.

Мы стараемся предоставить вам как можно больше важных вспомогательных файлов. В большинстве случаев, последние версии библиотек и другие пакеты мы включаем в каталог «Essentials» (Главное) на прилагаемом диске. Поэтому, если в вашей системе возникли проблемы с зависимостями, следует заглянуть именно туда.

ФОРМАТЫ ПАКЕТОВ

Мы стараемся включать как можно больше различных типов установочных пакетов: RPM, Deb или любые другие. Просим вас принять во внимание, что мы ограничены свободным пространством и доступными бинарными выпусками программ. По возможности, мы будем включать исходные тексты для любого пакета, чтобы вы смогли собрать его самостоятельно.

ДОКУМЕНТАЦИЯ

На диске вы сможете найти всю необходимую информацию о том, как устанавливать и использовать некоторые программы. Пожалуйста, не забывайте, что большинство программ поставляются вместе со своей документацией, поэтому дополнительные материалы и файлы находятся в соответствующих директориях.

ЧТО ЭТО ЗА ФАЙЛЫ?

Если вы новичок в Linux, вас может смутить изобилие различных файлов и расширений. Так как мы стараемся собрать как можно больше вариантов пакетов для обеспечения совместимости, в одном каталоге часто находятся два или три файла для различных версий Linux, различных архитектур, исходные тексты и откомпилированные пакеты. Чтобы определить, какой именно файл вам нужен, необходимо обратить внимание на его имя или расширение:

имя_программы-1.0.1.i386.rpm – вероятно, это бинарный пакет RPM, предназначенный для работы на системах x86;

имя_программы-1.0.1.i386.deb – такой же пакет, но уже для Debian;

имя_программы-1.0.1.tar.gz – обычно это исходный код;

имя_программы-1.0.1.tgz – тот же файл, что и выше по списку: «tgz» - это сокращение от «tar.gz»;

имя_программы-1.0.1.tar.bz2 – тот же файл, но сжатый bzip2 вместо обычного gzip;

имя_программы-1.0.1.src.rpm – также исходный код, но поставляемый как RPM-пакет для упрощения процесса установки;

имя_программы-1.0.1.i386.FC4.RPM – бинарный пакет RPM для x86, предназначенный специально для операционной системы Fedora Core 4;

имя_программы-1.0.1.ppc.Suse9.rpm – бинарный пакет RPM, предназначенный специально для операционной системы SUSE 9.x PPC;

имя_программы-devel-1.0.1.i386.rpm – версия для разработчиков.

Если диск не читается...

Это маловероятно, но если все же прилагаемый к журналу диск поврежден, пожалуйста, свяжитесь с нашей службой поддержки по электронной почте: disks@linuxformat.ru

LINUX FORMAT 4 GIGABYTE DVD

Декабрь 2007

LXFDVD99

Xubuntu 7.10

Ультра-быстрая разновидность Ubuntu – идеальна для старых машин.



Плюс:

» Linux From Scratch 6.3: Собери дистрибутив с нуля
» LMMS 0.3: Фантастическая мультимедиа-студия
» Ответы: Более 130 решенных проблем!
» Документация: HOWTO и руководства
» netPanzer: Сетевая тактическая игра

SmoothWall Express 3.0

Защитите свою сеть дистрибутивом-брандмауэром.

OpenOffice.org 2.3

Улучшенные диаграммы, ускоренные презентации и экспорт в MediaWiki.



LINUX FORMAT

Содержание LINUX FORMAT DVD Декабрь 2007

LINUX FORMAT

Сторона 1

Рабочий стол
GCompris - комплект обучающих программ
Leafpad - простой редактор на GTK+
LMMS - музыкальная студия
Pal - календарь для командной строки
Plan - планировщик
Sound Juicer - инструмент для снятия аудиотреков с дисков
Sunbird - это календарь-органайзер
Wyrd - текстовый интерфейс для Remind
X.org - графический сервер

Разработка
FreeBASIC - свободная реализация BASIC
Free Pascal - профессиональный компилятор языка Pascal
G95 - стабильный компилятор Fortran 95
GCC - коллекция компиляторов GNU
GCL - Common Lisp проекта GNU
GForth - быстрая и переносимая реализация ANS Forth
GHC - открытый компилятор Haskell
GNU Prolog - компилятор языка Prolog
Mono - свободная реализация .NET
MonoDevelop - среда разработки для Mono
Moscow ML - легковесная реализация SML
NASM - ассемблер для Intel x86
Caml - мультипарадигменный язык программирования
OpenCOBOL - компилятор языка COBOL
Perl - стабильный скриптовый язык программирования
Python - скриптовый объектно-ориентированный язык
Ruby - интерпретируемый объектно-ориентированный язык
Eiffel - чистый объектно-ориентированный язык
TinyCOBOL - компилятор COBOL

Дистрибутивы
DSL - миниатюрный Linux на флешке
Feather Linux - дистрибутив для работы с компакт-диска или USB-носителя
LFS - соберите свой дистрибутив!
SmoothWall Express - открытый межсетевой экран на базе GNU/Linux
Xubuntu - легковесная вариация Ubuntu

Игры
Blockout - трёхмерный тетрис
Bubbs Brothers - многопользовательский клон Bubble Bobble
FreeCol - свободная версия стратегии Colonization
Mini_Tanques - танковые сражения
netPanzer - тактическая стратегия
Powermanga - аркадная двумерная стрелялка
SDL Asylum - порт компьютерной игры Asylum

Справка
Ответы на часто задаваемые вопросы
Rute - книга по администрированию Linux

Hot Picks
CellWriter - панель для рукописного ввода текста
DigiBand - домашняя версия симулятора гитары и ударных
Evince - просмотрщик документов
Firewall Builder - инструмент для настройки межсетевых экранов
Highlight - универсальный конвертер исходных кодов
MikeOS - операционная система на ассемблере
Open Vovie Editor - простой редактор видео
Transcode - утилита обработки видео
Widelands - открытая стратегия реального времени

Интернет
Epiphany - веб-браузер для среды Gnome
Evolution - почтовый клиент
Adobe Flash Player - плагин для веб-браузера
SeaMonkey - комплекс Интернет-приложений
Willow - это прокси-сервер

Офис
AbiWord - текстовый процессор
Gnumeric - электронные таблицы
OpenOffice.org - комплект офисных программ

Поставляется вместе с журналом LINUXFORMAT номер 12(99) Декабрь 2007

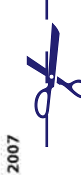
Сторона 2

ASP Linux 12

Комментарии? Присылайте ваши мысли и предложения по электронной почте: info@linuxformat.ru
Пожалуйста, ознакомьтесь с опубликованной в журнале инструкцией перед использованием данного диска.
Настоящий диск тщательно тестировался и проверялся на всех стадиях производства, однако, как и в случае с любым новым ПО, мы рекомендуем вам использовать антивирусный сканер. Мы также рекомендуем всегда иметь под рукой актуальную резервную копию данных вашего жесткого диска. К сожалению, редакция Linux Format не может принимать на себя ответственность за любые повреждения, разрушения или иные убытки, которые может повлечь за собой использование этого DVD, представленных на нем программ или данных. Перед тем, как устанавливать какое-либо ПО на компьютер, подключенный к сети, проконсультируйтесь с сетевым администратором.

Дефектные диски. В маловероятном случае обнаружения дефектов на данном диске, пожалуйста, обращайтесь по адресу: disks@linuxformat.ru

Тираж изготовлен ООО «Фортмедиа», Россия, Санкт-Петербург, 196006 ул. Цветочная д. 7, тел. +7 (812) 388-8290. Лицензия МПТР России ВАФ № 77-225



СОЗДАНИЕ УСТАНОВОЧНЫХ ДИСКОВ ПРИ ПОМОЩИ CDRECORD

Самый быстрый способ записать ISO-образ на чистую матрицу – это *cdrecord*. Для всех перечисленных ниже действий потребуются права root. Для начала определите путь к вашему устройству для записи дисков. Наберите следующую команду:

```
cdrecord -scanbus
```

После этого на экране терминала должен отобразиться список устройств, подключенных к вашей системе. SCSI-адрес каждого устройства представляет собой три числа в левой колонке, например, 0,3,0. Теперь вы можете с легкостью записать образ на диск:

```
cdrecord dev=0,3,0 -v /путь к образу/image.iso
```

Чтобы упростить дальнейшее использование *cdrecord*, сохраните некоторые настройки в файле **/etc/default/cdrecord**. Добавьте по одной строке для каждого устройства записи (вероятно, в вашей системе присутствует всего одно такое устройство):

```
Plextor= 0,3,0 12 16M
```

Первое слово в этой строке – это метка, затем, после адреса SCSI-устройства вы должны указать скорость и размер буфера. Теперь вы можете заменить SCSI-адрес в командной строке на выбранную вами метку. Все будет еще проще, если вы добавите следующее:

```
CDR_DEVICE=Plextor
```

Все, что вам теперь нужно для записи ISO-образа – это набрать команду

```
cdrecord -v /path/to/image.iso
```

Если вы не из числа любителей командной строки, в таком случае вам придет на помощь утилита *gcombust*. Запустите ее из-под root, выберите вкладку Burn и ISO 9660 Image в верхней части окна. Введите путь к образу, который вы хотите записать на диск, и смело нажимайте на Combust!. Пока ваш образ пишется на диск, можете выпить чашечку кофе.

Другая ОС?

Вам не обязательно использовать Linux для записи компакт-диска. Все необходимые файлы уже включены в ISO-образ. Программы вроде *cdrecord* просто переносят данные на чистую матрицу. Если у вас нет устройства для записи дисков, можно найти того, у кого оно есть, и записать диск на его компьютере. На нем может стоять Windows, Mac OS X, AmigaOS, или любая другая ОС.

Нет устройства для записи дисков?

А что, если у вас нет устройства, с помощью которого можно было записать образ на диск? Вы знаете кого-либо с таким устройством? Вам не обязательно использовать Linux для записи дисков, подойдет любая операционная система, способная распознать пишущий привод (см. выше).

Некоторые дистрибутивы умеют монтировать образы дисков и выполнять сетевую установку или даже установку с раздела жесткого диска. Конкретные методы, конечно, зависят от дистрибутива. За дополнительной информацией обращайтесь на web-сайт его разработчика. LXF

Содержание DVD

ЖУРНАЛ

JavaEE........Код примеров статьи.
Invite........Приглашение в LUG.
WavPack........Эффективный аудиокодек.
WordPress........Десятка лучших расширений.
wxWidgets........Исходный код утилиты Skype Monitor.

РАБОЧИЙ СТОЛ

GCompris........Пакет образовательных программ.
Leafpad........Легкий текстовый редактор.
LMMS........Программа для создания музыки.
Pal........Календарь командной строки.
Plan........Календарь и планировщик событий.
Sound Juicer........CD-риппер.
Sunbird........Календарь Mozilla.
Wyrd........Оболочка для Remind.
X.org........Графический сервер.

РАЗРАБОТКА

FreeBASIC........Компилятор BASIC.
Free Pascal........Компилятор Pascal.
G95........Компилятор Fortran 95.
GCC........Пакет компиляторов C/C++/Fortran/ADA.
GCL........GNU Common Lisp.
Gforth........GNU Forth.
GHC........Компилятор Haskell.
GNU Prolog........Компилятор Prolog.
Mono........Открытая реализация .NET.
MonoDevelop........Интегрированная среда разработки C#.
Moscow_ML........Реализация Standard ML.
NASM........Ассемблер x86.
OCaml........Объектный Caml.
OpenCOBOL........Компилятор COBOL.
Perl........Популярный скриптовый язык.
Python........Язык программирования.
Ruby........Динамический язык программирования.
SmartEiffel........Компилятор GNU Eiffel.
TinyCOBOL........Компилятор COBOL.

ДИСТРИБУТИВЫ

ASPLinux 12........Отечественный дистрибутив на базе Fedora Core (**СТОРОНА 2**).
DSL........Дистрибутив для старых ПК.
Feather........Дистрибутив для USB-брелка.
Linux From Scratch........Дистрибутив в стиле «сделай сам».
SmoothWall........Дистрибутив-брандмауэр.
Xubuntu........Шустрая разновидность Ubuntu.

ИГРЫ

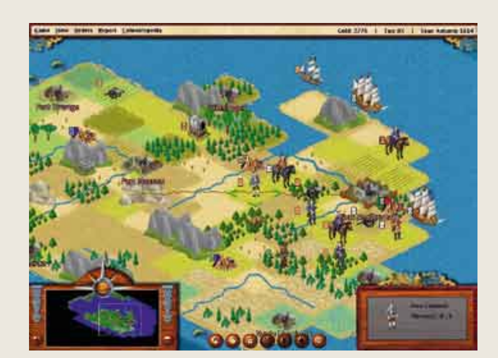
Blockout II........Трехмерное подобие Тетриса.
Bubs Brothers........Клон Bubble Bobble.
FreeCol........Клон Colonization.
Mini Tanques........Игра в стиле Battle City.
netPanzer........Тактическая военная игра.
Powermanga........Двумерная стрелялка.
SDL Asylum........Платформенная стрелялка.

HOT _ PICKS

CellWriter........Распознавание рукописного ввода.
Cryosleep........Генератор мозговых волн.
DigiBand........Имитатор гитары/барабана.
Evince........Программа для чтения PDF.
FWBuilder........Оболочка для брандмауэра.
Highlight........Программа для «причесывания» исходного кода.
MikeOS........Простая для изучения ОС.
OpenMovieEditor........Простой видеоредактор.
Transcode........Инструмент обработки потокового видео.
Widelands........Стратегия в реальном времени.

ИНТЕРНЕТ

Epiphany........Web-браузер.
Evolution........Календарь и клиент электронной почты.
Flash Player........Модуль расширения Adobe Flash.
SeaMonkey........Браузер и клиент электронной почты.
Willow........Прокси с фильтром контента.



› Колонизируйте Новый Свет в *FreeCol*, клоне классической игры от Сида Майера с открытым кодом.

ОФИС

AbiWord........Текстовый редактор.
Gnumeric........Электронные таблицы.
OpenOffice.org........Офисный пакет.

СПРАВКА

Answers........Решенные проблемы Linux.
RUTE........Руководство по
........администрированию Linux.

ГЛАВНОЕ

ATI driver........Графический драйвер.
Bash........Оболочка командной строки.
Cairo........Библиотека двумерной графики.
CheckInstall........Программа создания бинарных пакетов.
Coreutils........Утилиты командной строки.
GLib........Низкоуровневая библиотека.
Glibc........Библиотека GNU C.
Gtk........Инструментарий пользовательского интерфейса.
HardInfo........Системная информация и сравнительный анализ.
Jigdo........Создатель ISO-образов.
Kernel........Свежий релиз ядра Linux.
LibXML2........XML-анализатор и набор инструментов.
Ncurses........Оконный инструментарий текстового режима.
NVIDIA driver........Графический драйвер.
RAWRITE........Программа записи образов на дискеты.
SBM........The Smart Boot Manager.
SDL........Библиотека мультимедиа.



› *Epiphany*: стандартный браузер для Gnome. Движок на базе WebKit – в разработке!

LXF HotPicks

Ник Вейч
Чудо-математик, фанат Flickr и иногда редактор LXF, Ник рассылает из своего секретного бункера летучих роботов-обезьян на разведку интересных приложений в поле свободного ПО.

В ЭТОТ РАЗ ТОЛЬКО ДЛЯ ВАС: Cryosleep » Highlight » MikeOS 1.0.0 » Transcode » Widelands » Open Movie Editor » CellWriter » Evince » Firewallbuilder

Генератор шума окружающей среды

Cryosleep

Версия 0.2 **Сайт** http://cryposleep.yellowcouch.org

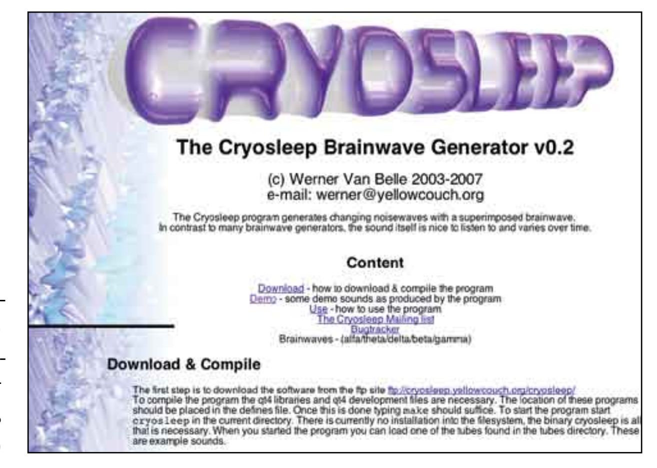
Я все еще мечтаю об Органоне, вместе с певицей Кейт Буш. Но если мечты идут вразрез с их реализацией, почему бы не исправить ход вещей, подпустив случайного шума?

Люди с давних времен вырабатывают случайный «белый» шум и даже «розовый» шум, якобы воздействуя на мозг (речь не о мучительной головной боли). Эти названия идут от распределения мощности шума по частотам. У белого шума частотный спектр равномерный. Громкий белый шум звучит довольно жестко и неприятно для ушей – он похож на статический шум ненастроенного телевизора (из тех, что не подавляют этот шум автоматически). Розовый шум подчиняется формуле p/f, то есть он тише на высоких частотах, и звучит более естественно, напоминая грохот прибоя на каменистом пляже. Люди придумали обозначать частотные спектры шумов различными цветами.

Цель *Cryosleep* – издавать шум, но особым образом. Здесь используются низкочастотные осцилляторы (LFO) для генерации псевдослучайного шума; формой их волн можно управлять с помощью верхней и нижней частоты отсечки и простого коэффициента. И вот что интересно: *Cryosleep* может накладывать «мозговые ритмы» на эти звуки. Говорят, что разные частоты воздействуют на мозг различными способами, повышая творческие способности, освобождая от тревожности и т.д. Если не в меру на этом зацикливаться, можно достичь и обратного эффекта...

Cryosleep требует для компиляции Qt4, но косвенно зависит также от *Qt3*, что может вызвать осложнения (совместимость *Qt4* различается в разных дистрибутивах, наборах патчей и т.д.). При желании им заняться, скачайте последнюю двоичную версию с сайта – это довольно просто, и должно работать на любой сборке Linux.

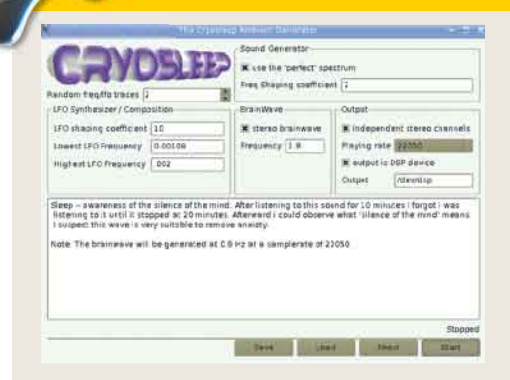
> «Говорят, что разные частоты влияют на мозг по-разному.»



» Сайт Cryosleep располагает обширной информацией и множеством вариантов настроек.

В директории **tubes** хранятся предустановленные звуки, дающие представление о возможностях. Запустите один из них, поставьте галочку в правом нижнем углу для выбора dsp-устройства вывода и укажите путь к нему (обычно **/dev/dsp**) в текстовом поле. Теперь подключите усилитель, крутаните уровень громкости до отказа и нажмите Start. Когда вы поставите мебель на место и распугаете всех позвоночных в радиусе 200 м, попробуйте все-таки разобраться с настройкой.

Шаг за шагом: Создаем аудио, таки съедающее мозг



» 1. Расслабление кишечника
Выберите параметры **10, 0.00108, 0.002** при **2** LFO-следах и включите мозговой стереоритм с частотой **1.8** и «идеальный» (perfect) спектр.



» 2. Деформация гиперкосмоса
Попробуйте **0.65, 0.00108, 0.8** при **10** LFO-следах, частоту мозгового стереоритма **3.6**, используйте «идеальный» спектр. Все как у Бака Роджерса!



» 3. Болезненный дискомфорт
Попробуйте **12** LFO-следов и параметры **0.85, 0.108, 12**, коэффициент частотного шейпинга **.2** и частоту мозгового стереоритма **10.5**. Вызовите врача.

Утилита для кодировщиков

Highlight

Версия 2.6.4 **Сайт** www.andre-simon.de/doku/highlight/en/highlight.html

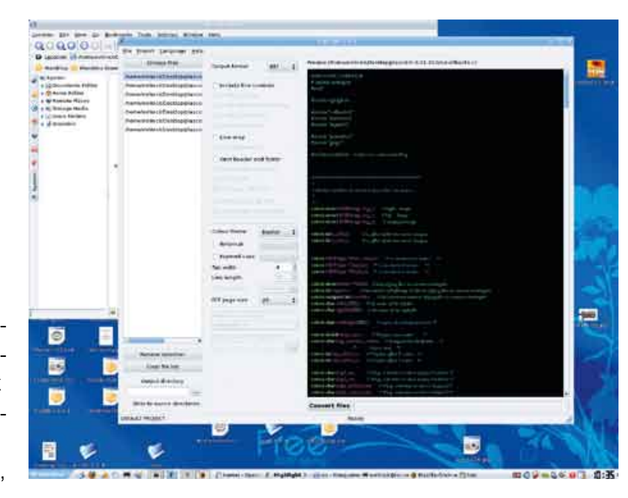
Ощущали ли вы нехватку идей насчет того, какой хороший проект с открытым кодом порекомендовать непростым дядюшке или тетушке (они ведь такие, «программисты»? – ред.) на Рождество? Мы и тут выкатим решение: утилиту, особенно ценную для программистов, которым приходится пояснять свой код в презентациях, книгах, на сайтах или как-то еще.

В целом, эта утилита улучшает вид кода. Вы запихиваете в нее жутко неорганизованные, корявые строки вашего последнего творения, и программа сама преобразует их в файл, который выглядит как экспонат выставки. На выходе ваш код может быть сохранен в одном из восьми различных форматов, в том числе XML, HTML, XHTML, TeX, RTF и простой текст (подробнее об этом позже). Это означает, что код можно раскрасить и легко отобразить на web-странице (web-форматы включают CSS-файл, для пущего лоска).

Поддерживаются разные стили форматирования, в зависимости от языка ввода/вывода. Код на C, например, можно привести к стилю ANSI (фигурные скобки с новой строки) или стилю учебника Кернигана и Ричи (фигурные скобки в конце объявления), или, если надо, к стилю Linux, GNU или даже Java. Если вы преобразуете код в текст, то просто получите опрятный его вид, без выделения синтаксиса.

Можете использовать командную строку с заданными опциями, или вообще создать скрипт для вашего текстового редактора, применяемый автоматически. Иногда вы не знаете заранее, что вам понравится – тут на помощь придет GUI-версия *Highlight*.

> «Независимо от языка программирования, ваш код станет опрятным.»



› Даже самый неряшливый код в *Highlight* способен стать элегантным.

Загрузите ваш файл (да хоть и целую группу файлов, если угодно), затем выберите параметры вывода в панели посередине. Вы увидите, что правая сторона дисплея выдает прокручиваемый предпросмотр кода на выходе.

Для web-руководств – вещь исключительно полезная, потому что можно добавлять собственные пояснения между кусками кода (а я так вижу, что это годится и для книг).

Уродливому на вид и трудно читаемому коду больше нет оправданий: ведь приведение его в порядок не требует от вас особых усилий.

Операционная система

MikeOS 1.0.0

Версия 1.0.0 **Сайт** http://mikeos.sourceforge.net/

Вы наслышаны об обезьянах, пишущих машинках и полном собрании сочинений Шекспира? А вот предлагаемый опус вполне мог бы быть ранним опытом небольшого подмножества обезьян, сроду не читавших инструкций. MikeOS, по сути, отнюдь не великий труд литературного гения. Она не литературна, не гениальна и не велика, а если честно, то и ярлык «труд» для нее слишком пышный. На самом деле это законченная* операционная система для архитектуры x86.

Возможно ли, чтобы один человек, при любом безрассудстве (в смысле, таланте?), создал всю ОС с нуля? В том-то и штука. Даже не заостряя на том, что Майк не гений и прочее, это не так сложно, как кажется. MikeOS изготовлена, чтобы показать, как работает простая ОС для x86, и интенсивно использует BIOS для поддержки ввода-вывода. Если вы хотите добраться до самого скелета ОС, или заинтересованы в изучении архитектуры x86, это очень хорошая стартовая площадка. Попробовать-то не больно! Вся ОС помещается на дискете, если вы и вправду хотите её установить; но, возможно, целесообразнее создать для прогона кода эмулятор (особенно если вы собираетесь поиграть с ним). *Qemu* прекрасно подходит для этой цели, а если вы заглянете на главную страницу проекта на SourceForge, то найдете вполне дружелюбную документацию, которая поможет вам насладиться преимуществами 16-разрядной ОС...

Из-под нее даже запускаются программы! Благодаря элементарной поддержке DOS и файловой системы FAT, можно раскрутить некоторые древние DOS-игры на небывалые результаты.

В документации также содержится много рекомендаций по ассемблеру x86 и созданию собственной ОС.

* то ли да, то ли нет.

> «Если 32-битный ПК для вас чересчур шикарно, то вот самая свежая альтернатива.»



› Вы тоже можете спустить всю мощь вашего ПК с поводка!



› А вы купили бы у этого человека б/у ОС?

Конвертор видеофайлов

Transcode

Версия 1.0.4 **Сайт** www.transcoding.org

Мир часто склонен неприятно нас озадачивать. Мучаясь жаждой, вы можете оказаться, например, у автомата с напитками. Вроде не так все и худо? Ан, худо, если автомат настроен на евро-центы, а у вас одни фунты.

Различия файловых форматов могут вызвать похожие проблемы. Многие из современных мобильных устройств, чтобы поберечь процессор и память, поддерживают ничтожно малое количество форматов видео, причем некоторые из этих форматов настолько непостижимы, что даже безупречно совместимый VLC будет тщетно пытаться приставить к делу их кодеки, построенные на базе квадратного корня из минус единицы. Оно бы тоже не беда, но ведь различаются даже форматы у моделей от одного изготовителя! Форматов на выбор гораздо больше, чем выходных в календаре члена Европарламента, и волей-неволей приходится задуматься о переводе из одного формата в другой.

Transcode достаточно давно удерживает лидерство среди приложений, решающих данную задачу. Его охват всяческих видео и аудио форматов расширяется по ходу борьбы с их Вавилонским смешением, что достигается, в частности, перекладыванием работы на другие библиотеки и другие утилиты, вроде *ffmpeg*; но *Transcode* всех их объединяет в одной, простой для понимания командной строке. Стоп: я сказал, «простой для понимания»?..

Transcode Command Line Options



Transcode accepts a lot of commandline options, which are passed on to the various **tc*** binaries while doing its work.
(check the man pages of the other **tc*** binaries if an option's explanation isn't clearly described here) To see differences in commandline options between releases, see commandline options changes

Default values are shown between square brackets and in [green].

Command line in principle

This is a basic command line for most uses

```
transcode -i sourcevideo \
          -p sourceaudio \
          -o destvideo   \
          -x vmod[,amod] \
          -y vmod[,amod] \
          -J filter ...
```

The **-i** option specifies the filename of the source video.
The **-p** option comes in handy if the video file contains no audio stream, or if you prefer to use a separate audiofile.
The **-o** option sets the name of the output file. If omitted, transcode will encode to '*/dev/null*'.
The **-x** option sets the video[,audio] import modules. If omitted, transcode will probe for appropriate import modules.
The **-y** option sets the video[,audio] export modules. If omitted, transcode will ignore the **-o** option and encode to '*/dev/null*'.
The **-J** option is for filter plugins.

The order you use to specify the options is not important.
(also available: the man page for transcode)

Input

```
-i name
   input file|directory|device|mountpoint which can be
   an .avi, .mkv, .ogm, .vob, .mpg, a dvd device...
   nearly any type of video/audio container, device or
   file type.

-H n (0=off) [1]
   auto-probe n MB of source file specified using the
   -i option. Setting to '0' will prevent the auto-probe
   from overriding geometry specified by the -g option,
   for example.
   if you use -H 0 you will need to set
   --export_fps and --export_asr and
```

Использование программы

К несчастью для многих, за мощь *Transcode* надо платить, и ценой является сложность. *Transcode* – одно из немногих приложений в мире, которому нехватает буквенно-цифровых символов для ключей командной строки (даже после разделения на символы верхнего и нижнего регистров). В итоге какая-нибудь операция конвертации **mpeg** в **mp4**, которую вы воображали простой, может резво обернуться целым экраном буквенной мешанины. Чтение man-страниц очень часто не выход, и не потому, что они плохо написаны: там столько информации, что трудно найти волшебные опции, которые вам нужны.

Опять же, это не потому, что программа плохо написана, а потому, что мир видео достаточно сложен и запутан. Во-первых, существуют разные частоты кадров. Ваша дешевая цифровая камера может претендовать на съемки видео – пусть даже и неплохого, но она наверняка делает 15, может 20, кадров в секунду. На компьютере смотрится вполне прилично, сойдет также для отправки электронной почтой вашей бабуле, но при попытке проиграть его через телевизор или создать видеодиск начнутся проблемы. Системы PAL работают с 25 кадрами видео в секунду и бывают очень придирчивы – некоторые DVD/CD-плейеры впадут в кому, если им подсунуть видео с другой частотой кадров. И это только один аспект видеофайла – а есть ведь и разрешение, и соотношение сторон кадра; все эти параметры надо иметь в виду. Причем мы пока обсуждали физические характеристики видео, не затрагивая методов кодирования и их опций. Кстати, еще есть звук…

Рецепты

Как видите, задача сложнее, чем казалось на первый взгляд. Без сомнения, есть эксперты, которые помнят все ключи команды, но нам они пока не встречались. Большинство людей полагаются на проверенные рецепты для своих частных задач, при необходимости подправляя их.

Мы – ребята хорошие, и поделимся с вами некоторыми рецептами, которые могут быть полезны. В простейшем случае *Transcode* надо указать входные и выходные имена файлов и, конечно же, метод кодировки:

› Все опции запомнить невозможно, так что не гнушайтесь обращаться за помощью.

```
transcode -i mysource.avi -y xvid4 –o
outfile.avi
```

Ключ y управляет модулем экспорта для генерации выходного файла. В данном случае мы создаем видео-объект **mpeg4**, используя библиотеку *xvid*. Хотя мы здесь и указали тип входного файла, делать это не обязательно, потому что *Transcode* автоматически проверит файл на детали формата. В следующем примере попытаемся изменить еще и размер видео. Без этого не обойтись, например, если нужно сделать DVD, способный воспроизводиться на любом плейере, или вогнать видео в камкордер.

```
transcode -i mysource.avi -y dvraw –Z
720x576 -o outfile.dv
```

Ключ -Z активирует высококачественный модуль изменения размера видео: он приводит кадры к правильному размеру. Размер указан для разрешений PAL DVD/DV, в NTSC он другой.

HotGames Развлекательные приложения

Стратегия

Widelands

Версия Build11 **Сайт** http://xoops.widelands.org/



Linux Format Hottest Pick

В давние времена, когда люди в индустрии игр еще пытались создавать новые и волнующие жанры, компания Bluebyte добилась неожиданного успеха, выпустив игру *Settlers* – в общем, подобную всем другим играм, где нужно было управлять ресурсами, но довольно причудливую и необычную. Все работало на микроуровне, а главным были дороги, используемые при постройке поселения. Практически все вертелось вокруг троп и умении вашего малого народца хватать и носить.

Widelands пыталась перенять этот стиль игр под Linux. Наверно, будет преувеличением сказать «и обновила их под наш век», потому что игра очень и очень напоминает оригинал.

Забота стратегий этого типа – «эффект бутерброда с беконом», и данная игра её тоже не избежала. Сценарий такой: шахтеры должны быть сыты, чтобы они могли работать. Сырье, которое они производят, нужно для строительства зданий, где производится еда. Высший уровень развития общества в игре – это не возможность производить реактивные истребители или колесницы, а способность штамповать полосатые бутерброды, чтобы удовлетворить спрос. Организуйте это, и все станет возможно.

Это подвиг не из легких. Вам понадобятся хлеб и бекон. Бекон поступает со свинофермы, для строительства которой нужны доски и камни. Доски делают на лесопилке, но туда нужно доставить лес. Камень можно добыть на каменоломне, а если открытый карьер исчерпан, остается рыть шахту. Таким образом обеспечивается поставка сырья. Хлеб пекут на хлебозаводе (снова камни и доски), для него нужны мука с мельницы (опять камни и доски) и вода… Вот так все и работает. В целом, ради бутерброда нужно построить город довольно большого размера. А потом, естественно, ленивые шахтеры захотят попить пивка…

> «Игра-стратегия Linux: смотрите, как сжирают ваши продукты!»

› «Ну есть там золото на этих холмах, говорю вам». Новая эпопея Сьерра-Мадре…

Хотя это звучит достаточно пугающе, игра все-таки затягивает: непросто добиться эффективной расстановки зданий так, чтобы перевозки леса и прочих материалов не замедляли процесс. В игре есть несколько сценариев и пара кампаний, которые, в основном, построены на стремлении выполнить производственный план, но вдобавок есть и кое-какие боевые цели.

Конфликты культур

Борьба пока в достаточной мере не разработана. Хотя имеются возможность строительства военных зданий и оружия и способы тренировки армии, борьба сама по себе представляет собой кликанье на объекте и посылку людей атаковать его в надежде на лучшее.

Чтобы разнообразить игру, в нее включили концепцию различия культур. В настоящее время в игре доступны две расы: варвары и имперцы. У последних здания выглядят лучше, совершенно иная технология, и они считают себя более цивилизованными. Варвары, с другой стороны, строят более убогие здания, и их счета за уборку значительно ниже.

Игра довольно интересна, но в ней есть ряд нервирующих моментов: например, шахтеры иногда норовят истребить все бутерброды с беконом и осушить запасы эля, ничего не производя. Уж если такое случается, самое лучшее – в стиле Маргарет Тэтчер – закрыть шахту и выстроить другую.

Мы очень редко даем игре звание *Hottest Pick*, но она обещает многое, с легким привкусом ретро. Попробуйте!

Исследуя интерфейс



Трон королей
Варварский штаб выстроен без изысков, но впечатляет.

Работа по добыче угля
Шахты поставляют сырье, а шахтеров надо кормить.

Угнетение народных масс
Все тяжести поднимают и перетаскивают шахтеры.

Есть, босс!
Приглядывайте за производством: прирост – это деликатный вопрос баланса.

Капитанский мостик
Морское богатство Нептуна: рыбные палочки к чаю.

Эффект желтой прессы
В лесах чужаки! Приходят, работу нашу забирают!

Редактор видеофильмов

Open Movie Editor

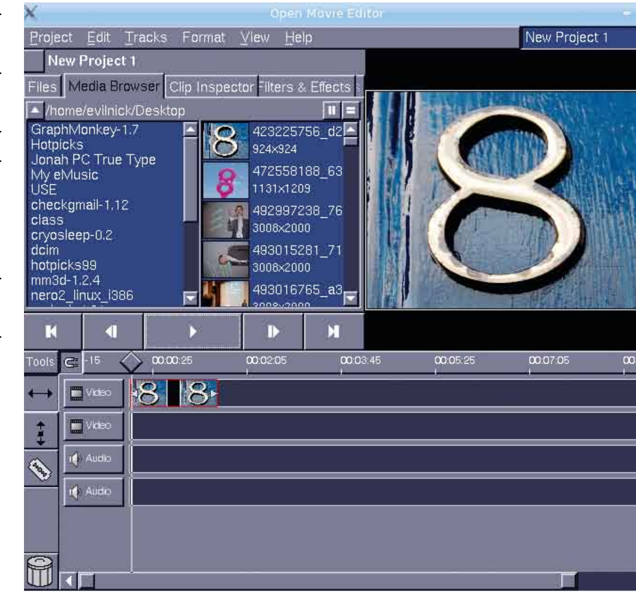
Версия 0.0.20070917 **Сайт** http://openmovieeditor.sourceforge.net

Странно, что мы добавили его в HotPicks, но у нас есть веская причина для этого. Странно – потому что, при несомненной перспективности, программа пока не особенно хороша. Как вы могли догадаться по названию, это приложение для обработки ваших видеофильмов. Оно достаточно ограниченное: редактировать можно только файлы *Quicktime*, **.mov**. Да и то с оговорками, потому что даже с файлами этого формата бывают проблемы.

Возможно, не совсем справедливо, выбрав приложение, тыкать в его недостатки: существует бездна намного худших, а в этом приложении заложен реальный потенциал. Его недостатки скорее символизируют неспособность сообщества Open Source в целом создать достойный редактор видео. *Kino* хорош для фильмов, скачанных прямо с DV-камеры, но для другого использования там слишком много ограничений. Альтернативный вариант – перегруженная наворотами *Cinelerra*, но, несомненно, существует спрос на приложения и такого типа – простой редактор видеофайлов. Он мог бы даже переломить ситуацию, при обратной связи от людей, которые помогли бы добавить функциональности.

Так или иначе, в программе имеются некоторые базовые эффекты и настройки, возможность загружать **.mov**-файлы и статические JPEG, добавлять названия, играть со звуком, и все это – через дружелюбный интерфейс. Все функции доступны перетаскиванием, монтажный стол внизу должен быть понятен всем, кто когда-либо работал в подобных приложениях на других платформах. Если программу немного доработать и дополнить количество форматов, она может стать действительно полезной маленькой утилитой.

> «Подключайтесь: это приложение имеет потенциал.»



› Отличный старт в довольно неразвитой области приложений Linux.

Распознавание символов

CellWriter

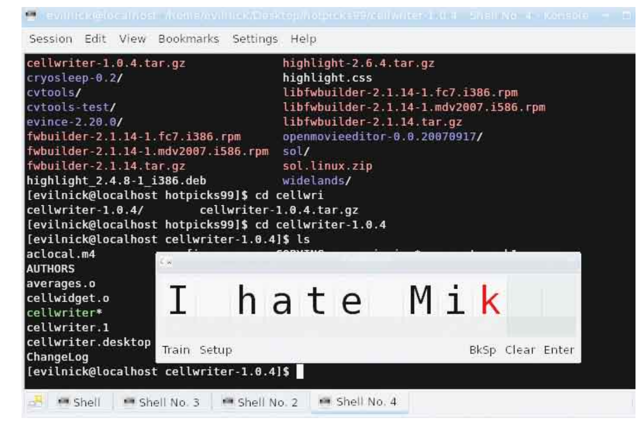
Версия 1.0.2 **Сайт** http://risujin.org/cellwriter/

После ухода фрейлейн Смолли из Башен LXF, средняя разборчивость почерка членов команды резко снизилась. Мало того, что большинство образчиков записей выглядят как творения паука, принявшего чернильную ванну – паук должен дополнительно заблудиться, напиться допьяна и обуть специфические ботинки. И когда кто-то у нас экспериментирует с каким бы то ни было приложением для распознавания рукописного текста, это служит поводом ко всеобщему веселью: если даже шибко умные сотрудники не в силах расшифровать надпись на бумажном пакете, у программного продукта шансов теоретически нет.

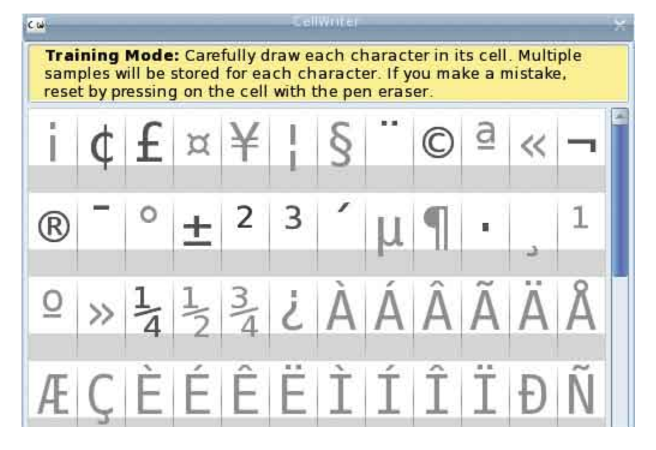
А вот *CellWriter* справляется. Имейте в виду, что это не ПО общего назначения: оно было создано для использования во встроенных приложениях – планшетных устройствах, телефонах и так далее. Программа следует подходу «пишите слова в этой строке в этом окне», который намного облегчает декодирование. И все же результаты впечатляют. Она явно работает гораздо лучше при использовании графического планшета для ввода информации, но точность распознавания поразительная, даже при рисовании символов мышью. При первом запуске программы вам будет предложен тяжкий тренинг. Ну, мы только хотим сказать, что писание, в отличие от печатания, считается в наше время трудным.

По сути, вы должны заготовить формы для всех символов, которые вы когда-либо собираетесь использовать. В отличие от аналогичного программного обеспечения, вам нужно только прописать каждый символ по разу (вы можете повторить процедуру, если допустили ошибку), а затем переключиться на расширенный набор символов, чтобы добавить экзотических начертаний.

В чем же смысл? В активном режиме можно писать в клетках на экране и нажимать Enter. Вы видите добавление символов на экране, а после нажатия Enter весь текст передается в соответствующее активное приложение – так можно общаться с любой программой, требующей ввода текста. Пользы тут больше для встраиваемых приложений, но, безусловно, попробовать интересно.



› С планшетом довольно легко писать красивым почерком.



› Тренинг может показаться тяжким, особенно с учетом количества специальных символов.

Просмотрщик документов

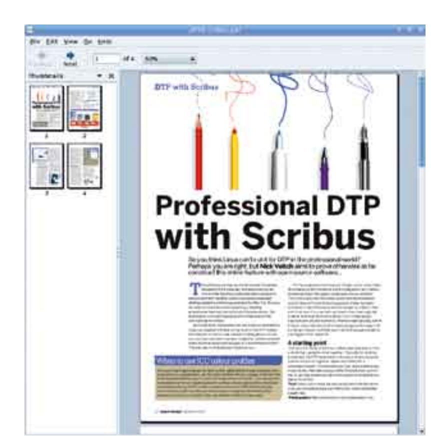
Evince

Версия 2.19.92 **Сайт** www.gnome.org

У вас, возможно, уже стоит *Evince* – программа, которую породило стремление создать простое приложение для просмотра документов различных форматов с должным отображением на экране, чтобы сократить число вспомогательных приложений, требующихся для этой цели при использовании Gnome.

Базовые форматы *Evince* – PDF и PostScript (используя *poppler* и *ghostscript* соответственно). Так как PDF – самая распространенная форма документации после «сырого» текста и HTML-файлов, важно, чтобы она отображалась правильно. Linux-версия *Adobe Reader* разработана довольно слабо, и в некоторых релизах изрядно тормозит и сбоит, так что неплохо иметь программу, на которую можно положиться.

DjVu-файлы также поддерживаются. Для непосвященных, это формат хранения документов, принятый в фирме AT&T. Многие архивы были отсканированы и хранятся именно в этом формате, так что он заслуживает внимания. Поддерживаются и файлы **.dvi** (не зависящий от устройства формат *TeX*), а также идет много дискуссий о других типовых форматах файлов, с которыми должна или не должна работать программа. Например, некоторые ныне поддерживаемые графические форматы могут в будущем исчезнуть из ее списка, поскольку разработчики озадачены попытками заставить *Evince* делать слишком много вещей помимо обработки документов. Не хватает поддержки форматов для «комиксов»: ведь **.cbr**, **.cbt** и **.cbz** – это просто архивы JPEG- или PNG-изображений, хранящиеся в последовательности (последняя буква расширения указывает на метод сжатия: *rar*, *tar* или *zip*). *Evince* проста на вид, но фактически это сила.



› *Evince* за последний год стала работать намного быстрее: попробуйте, даже если вы фанат KDE.

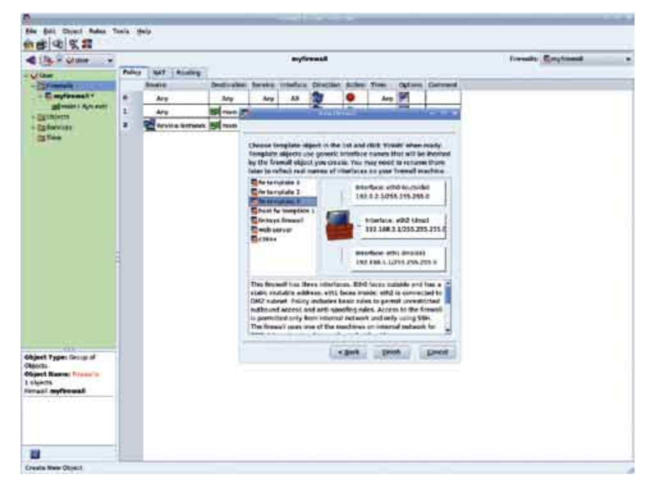
Сетевая безопасность

Firewallbuilder

Версия 2.1.14 **Сайт** www.fwbuilder.org

Firewallbuilder обновился с момента своего неудачного выступления в Сравнении в LXF98, хотя не все критические замечания были отработаны. Тем не менее, это очень способная система. Программа обрабатывает все как объекты, будь то сетевой сервис, протокол, хост, сеть или отдельный интерфейс. Затем из объектов можно собрать правило, перетаскивая их из панели справа в его определение. Вам даже не придется самому мучиться с созданием объектов, потому что таковых включено немало: стандартные протоколы и параметры сети; есть даже ряд полных наборов для типовых сценариев.

› Для экономии усилий используйте сохраненные определения.



Что и как перетащить

Хотя создание брандмауэров при помощи перетаскивания выглядит простым, это не совсем так. Придется еще попотеть, составляя разумные правила, заставляя объекты работать нормально и приготовляя их к сборке. Этот последний момент связан с тем, что *FirewallBuilder* может выбирать цель для фактической компиляции и установки правил. Это может быть Linux-машина с iptables или *ipchains*, сервер FreeBSD или Mac OS X, или даже устройство Linksys или Cisco. Идея заключается в том, что даже не нужно запускать *FirewallBuilder* на хосте, для которого он предназначен.

Создав список правил, нужно скомпилировать и установить его. В комплекте с приложением идут некоторые помощники для создания из ваших правил файлов в удобном формате, но другим позитивным моментом является то, что после сохранения правил вы можете скомпилировать один и тот же набор для нескольких устройств, даже если они находятся на отдельных платформах! Готовые правила могут быть установлены с помощью *fwbuilder*, в зависимости от выбранной системы.

Имеется и руководство, но оно выглядит довольно поверхностным в одних аспектах и устаревшим в других, хотя и обновляется. Возможно, вы не выберете его для быстрого создания брандмауэра в вашей домашней системе, но если вы управляете множеством устройств и располагаете временем, чтобы разобраться в приложении, ваши труды окупятся. LXF

Также выпущены

Новые и обновленные программы, заслуживающие вашего внимания...

Ekiga 2.0.11 ПО для видеоконференций, ранее известное как *GnomeMeeting*, стало более переносимым. www.ekiga.org

PHP 5.2.4 В языке, на котором зиждется Web, исправлено более 120 ошибок. www.php.net

Awesome 1.2 Набирающий популярность альтернативный оконный менеджер – теперь с улучшенными плавающими окнами. http://awesome.naquadah.org

PyPE 2.8.7 Редактор для программистов на Python со всевозможными «наворотами». http://pype.sourceforge.net/index.shtml

Leafpad 0.8.12 В новой нестабильной версии текстового редактора для Gnome появились развитые функции печати. http://tarot.freeshell.org/leafpad

fpcbol 0.9.9 Рабочий стол для детей (то есть с большими иконками). http://sourceforge.net/projects/fpcbol

Galcon 1.3.2 Многопользовательская смесь стратегии и экшна галактических масштабов. Также для Windows и Mac! www.imitationpickles.org/galcon

Gajim 0.11.2 Jabber-клиент на базе Python/GTK. Исправлены некоторые досадные ошибки. www.gajim.org

Castget 1.0.0 Простая утилита для слежения и загрузки подкастов по RSS-лентам. www.nongnu.org/castget



Школы Санкт-Петербурга на особом положении

Галина Пожарина представляет проект www.spbschools.ru – свободное программное обеспечение для образовательных учреждений Санкт-Петербурга.



Информационные потоки, образующие основу управления учебным, воспитательным, организационно-административным и прочими процессами, составляющими суть деятельности учреждения образования, являются сложным контентом и, на современном этапе, требуют как понимания и структурирования их со стороны организаторов данных процессов, так и понимания и структурирования со стороны разработчиков, обслуживающих данные процессы информационных систем.

Слово «контент» уже достаточно прочно вошло в русский язык, по крайней мере, в интернет-сообществе. Начало активного употребления этого понятия связано именно с интернет-технологиями. "Content" переводится с английского как «содержимое», «содержание», «значение», «смысл». Достаточно часто в контексте интернет-технологий слово "content" переводят как «информационное наполнение».

Для оптимального проектирования и поддержки деятельности web-ресурса образовательного назначения должны быть задействованы специалисты трех направлений:

- Руководители информационной инфраструктуры образовательного учреждения (IT manager);
- Технические специалисты (technical support);
- Редакторский коллектив (content management).

На практике же эти разные специалисты могут быть представлены в школе в одном лице и даже не иметь достаточной технической подготовки. Сектор открытого программного обеспечения позволяет решить поставленные задачи с помощью систем управления динамическим контентом (CMS).

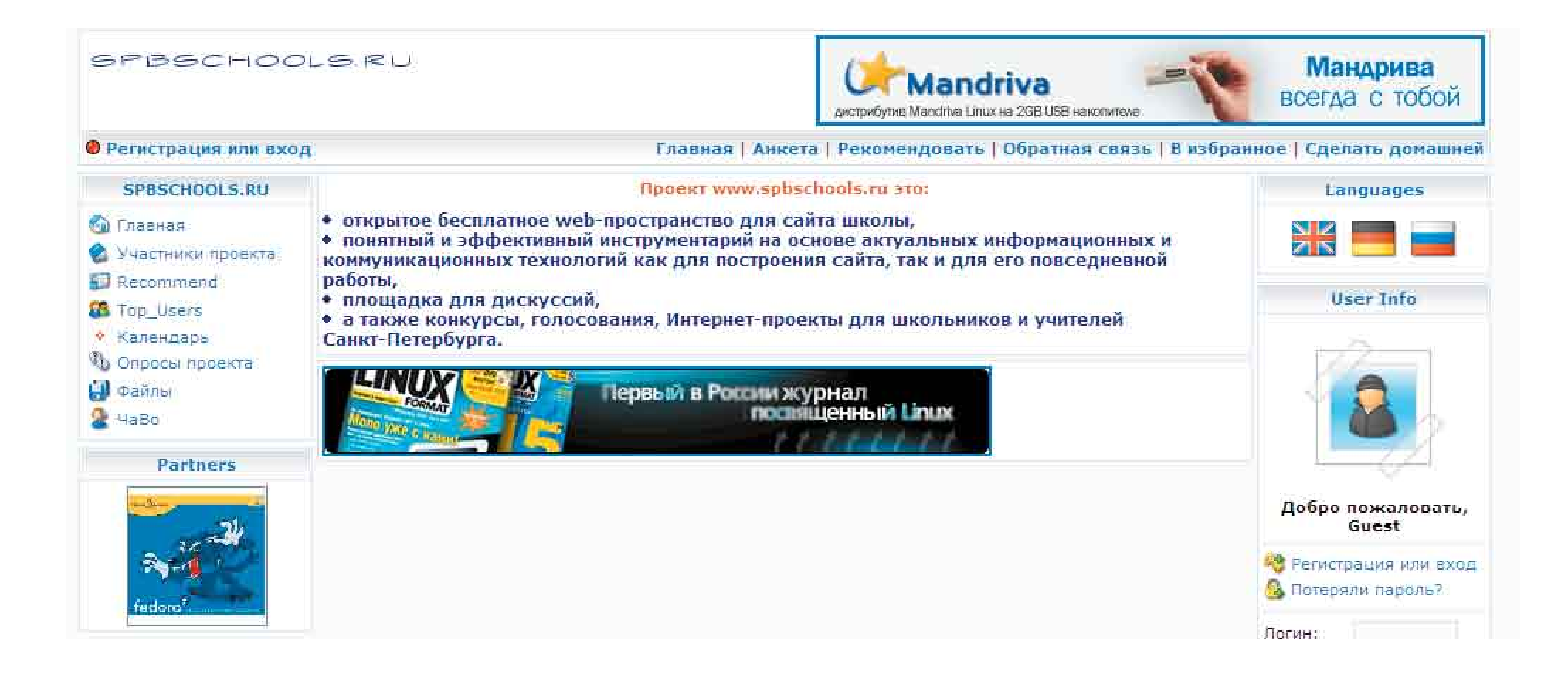
Система SLAED CMS написана на PHP и работает с базой данных MySQL. Это мощный программный пакет, позволяющий эффективно и гибко управлять интернет-проектами. На основе SLAED CMS любой желающий, даже не обладающий большими знаниями пользователь, может построить мощный портал.

Модульное наращивание позволяет инсталлировать разного рода модули, при помощи которых возможно дополнение стандартного пакета SLAED CMS. Систему можно использовать для любой языковой аудитории, а также одновременно применять на сайте неограниченное количество языковых версий. При наличии хотя бы небольшого опыта и знания HTML, вы сможете существенно изменить не только внешний вид, но и саму структуру сайта.

SLAED CMS является гибкой, мощной и интуитивно понятной системой с минимальными требованиями к хостингу, малым количеством запросов к базе данных. По оценкам некоторых специалистов, SLAED CMS является самым удачным выбором для построения современного ресурса.

В стандартный пакет системы входит большое количество модулей и функций, позволяющих как формировать контент ресурса, так и управлять работой авторов и пользователей.

На основе данного программного решения в Санкт-Петербурге реализуется проект www.spbschools.ru.



Проект www.spbschools.ru – это:

- открытое web-пространство для сайта школы,
- понятный и эффективный инструментарий на основе актуальных информационных и коммуникационных технологий как для построения сайта, так и для его повседневной работы,
- площадка для дискуссий,
- а также конкурсы, голосования, интернет-проекты для школьников и учителей Санкт-Петербурга.

Участники проекта получают бесплатную учетную запись для доступа к его ресурсам, а также техническое и методическое сопровождение для построения и поддержки web-ресурса школы. Для наполнения индивидуального ресурса школы достаточно уметь просто редактировать тексты. Техническое сопровождение и обучение редакторов ведут специалисты Центра образовательных инициатив Open Source Санкт-Петербурга.

Проект основан на использовании свободного программного обеспечения, предназначен для консолидации региональных образовательных ресурсов и уделяет приоритетное внимание как совершенствованию сетевого доступа к ресурсам учебных заведений, так и развитию «беспрепятственных» технологий на основе использования бесплатного, общедоступного информационного наполнения и открытых для всех пользователей программных средств, с соблюдением при этом прав на интеллектуальную собственность. LXF

«У самовара» с Александром Поносовым и ALT Linux



Алексей Евгеньевич Новодворский, заместитель генерального директора ALT Linux. В кругу пользователей Linux фигура, можно сказать, культовая. «Заказали» Алексея, а вернее интервью с ним мои «френды», пользователи LiveJournal, еще летом. Но встретиться удалось, только когда Алексей приехал в Пермь обсудить вопросы поддержки пользователей ALT Linux в Пермском крае со своим старым знакомым, единомышленником Алексеем Костаревым, директором ООО «Невод» (в центре на фотографии внизу). Эта фирма за последние 12 лет принимала самое активное участие в разработке ядра и программ для Linux, занималась внедрением свободных программ в Прикамье, то есть «готовила критическую массу».

Несмотря на плотный график переговоров, мне удалось задать Алексею Новодворскому несколько вопросов.

Александр Поносов (АП): **Первый вопрос от «френдов». С чего начинался ALT Linux?**

Алексей Новодворский (АН): Начинали еще в 1997 году, тогда мы с Алексеем Смирновым решили пропагандировать, популяризировать свободное ПО. До этого мы не были Unix'оидами, занимались распознаванием образов в разных вариантах, поиском логического вывода на самых разных платформах. Когда мы начали заниматься свободным ПО, то нас привлекла идеология. Очень многие приходят из Unix, а мы пришли именно по идеологическим соображениям. Мы издавали дистрибутивы, распространяли их. Debian, Mandrake, SuSE... Потом как производители стали вносить свои изменения. Вокруг стали появляться люди, которым это нравится. Когда подобралась сильная команда, решили создать свой бизнес.

На данный момент у нас большой коллектив, порядка 200 человек активных разработчиков, и в этом наша некоторая уникальная особенность. Это самый большой проект свободного ПО [в России, – прим. ред.].



АП: **Как директор директору: трудно управлять творческим коллективом?**

АН: Понятно, что такая команда создается на открытости, само собой, но и в значительной степени на харизме лидеров. Продавать дистрибутивы можно и без харизмы, а вот создавать проект, особенно большой и серьезный, без сильных лидеров нельзя.

Я восемь лет работал учителем, причем в самой сильной математической школе в Москве. То есть я отбирал ребятишек в свои классы – конкурс большой, и все такое прочее. Поэтому в какой-то степени я рассматриваю свою сегодняшнюю деятельность как продолжение той, в том плане, что я очень люблю работать с умными нестандартными людьми. И такие коллективы создаются, кроме всего прочего, на взаимном восхищении, когда каждый получает удовольствие от того, что работает в команде с настолько сильными людьми. Кто-то что-то сделает, им восхищаются, а он восхищается другими.

> *Такие коллективы создаются, кроме всего прочего, на взаимном восхищении, когда каждый получает удовольствие от того, что работает в команде с настолько сильными людьми.*
> *Алексей Новодворский*

АП: **Это как в басне: «За что кукушка хвалит петуха?»**

АН: На самом деле, нет, потому что это уважение, восхищение работой с одной стороны, с другой стороны – это всегда спор, в творчестве без споров не обходится. Так сделал, или не так, у каждого свои принципы. Вот это очень важно.

АП: **У ALT Linux солидное «собрание сочинений». Зачем понадобилось создавать собственный репозиторий? Может быть гораздо проще и выгоднее присоединиться к какому-либо крупному международному проекту?**

АН: Политика ALT состояла все эти годы в том, чтобы построить полный цикл разработки, независимый, а это очень длительный процесс,

причем нельзя сказать, что мы с самого начала ставили перед собой такую цель. Когда собираются сильные люди, то у них появляются свои варианты решений, которые не совпадают друг с другом, и появляются специфические задачи. Результаты решений этих задач хочется включить в репозиторий. Опять здесь наблюдаются расхождения. И даже на интернационализации, чем мы тогда занимались, у нас начались расхождения с Mandrake. Отсюда появился свой репозиторий. И если ты занимаешься разработкой, хочешь создать готовое решение на базе какого-то репозитория, то с неизбежностью ты приходишь к тому, что у тебя появляются расхождения с «родителем» и, в конце концов, самостоятельность. Наша эволюция была естественной, и она естественна для кого угодно. Все люди, которые честно делают свою работу и хотят вносить в репозиторий качественное решение, пройдут тот же путь.

Конечно, это большая проблема – найти прямой заказ на репозиторий, что практически невозможно. Прямых доходов наше главное направление не дает. Это значит, что мы все это время тем или иным способом зарабатывали деньги, для того чтобы «кормить» проект. Для меня Сизиф важнее, чем ALT Linux как бизнес-структура. Наша особенность – никакие мы не бизнесмены, и бизнес для нас – это средство для реализации проектов, подобных Сизифу.

АП: Каковы перспективы развития этого проекта?

АН: В России своя специфика. Я не вижу возможности сделать такой большой проект, как Debian, чтобы он не был ни при одной из фирм. Со временем это будет, но нужно, чтобы был достаточный уровень сытости людей, свободное время. Захотел поработать над проектом, поработал в свое удовольствие, на карточке денег хватает. Полгода поработал, деньги кончаются – пошел, заработал. И так далее. Если Сизиф со временем превратится в независимый проект, типа Debian, я буду только рад.

АП: В перспективе намечается массовое внедрение свободных программ в государственных учреждениях. Здесь уже не обойтись силами одной-двух компаний, необходима слаженная работа всех российских разработчиков. Какие, на Ваш взгляд, варианты сотрудничества возможны в дальнейшем?

АН: В качестве возможного проекта мы предлагаем следующее. В Сизифе можно собирать пакеты других репозиториев. Эту технологию отработал Виталий Липатов, глава Etersoft, они работают на Сизифе и в нем собирают Wine@Etersoft, который поддерживает около двадцати дистрибутивов. Технологии сборки у нас универсальны и удачны. Имея единый репозиторий, причем контролируемый своей командой, можно собирать дистрибутивы и тем фирмам, которые не имеют полного цикла разработки.

Кроме того, я уверен, что кроме своего репозитория в России должен быть национальный центр международной разработки свободного ПО. Чем глубже мы будем заниматься базовым уровнем, тем теснее будем взаимодействовать с международным сообществом.

АП: В последнее время все чаще идут разговоры о национальных дистрибутивах народов России и СНГ, что делается в этом направлении?

АН: У меня лично есть опыт локализации дистрибутива на белорусском языке [*смеется*] Мы делали и казахскую локализацию, и башкирскую, и киргизскую. Ничего сверхсложного нет.

Возьмем, к примеру, в вашем регионе коми-пермяков. Для таких народов делать локализацию нужно обязательно, чтобы они могли развивать свой язык. Здесь даже не нужно перевода интерфейса, а просто требуется, чтобы они могли набирать на нем тексты и печатать их. Тот язык, который не имеет компьютерной письменности, обречен на вымирание.

Для малых народов делать локализацию нужно обязательно, чтобы они могли развивать свой язык. Язык, который не имеет компьютерной письменности, ОБРЕЧЕН НА ВЫМИРАНИЕ.

Алексей Новодворский

Когда читатели LXF откроют этот номер журнала, уже будут объявлены итоги конкурса по внедрению пакета свободных программ в школах пилотных регионов. Мне очень хотелось расспросить Алексея об участии ALT Linux в этом конкурсе, но не стоит торопить – будем корректны. Надеюсь, что при любом результате, команда ALT Linux с прежней энергией будет продвигать свободные программы в образование, и что Сизиф построит для потомков свой Коринф. LXF

Котласские речники выбирают GNU/Linux

Современный мир – мир дешевых торговых путей, а самый дешевый из всех транспортов – водный. «Дешевый» вовсе не то же, что «примитивный» – это, скорее, значит «технически эффективный». Очень похоже на GNU/Linux. Linux Format беседует с **Дмитрием Владимировичем Моденовым** из Котласского речного училища.

Евгений М. Балдин, LXF: **Представьтесь, пожалуйста, Дмитрий Владимирович.**

Дмитрий Владимирович Моденов: Я являюсь заместителем директора Котласского речного училища по инновационным технологиям обучения, которое, кстати, и закончил в 1992 (тогда училище называлось Лимендским).

Высшее образование получил в Санкт-Петербургском государственном университете водных коммуникаций по специальности «Морское судовождение». Наше училище – филиал этого университета. Территориально мы расположены в городе Котлас, который является административным центром Котласского района Архангельской области.

Еще в студенчестве у меня появилась тяга к освоению компьютерной техники, которая со временем переросла из увлечения в работу. С апреля 2007 полностью работаю в Linux. На домашнем компьютере использую дистрибутив ALT Linux 4.0 Desktop, а на ноутбуке установлен Fedora 7.

LXF: **Кого готовит ваше училище?**

ДВМ: Котласское речное училище, филиал ФГОУ ВПО «Санкт-Петербургский государственный университет водных коммуникаций» (http://kotlasskoe.ru), ведет подготовку по восьми специальностям, пять из которых – технические. Остановлюсь поподробней именно на них, поскольку на сегодняшний день около 80% контингента учащихся обучаются на технических специальностях. Освоение технической специальности, на мой взгляд, сложнее, и связано это не только с возрастающими требованиями государственного образовательного стандарта к выпускнику и его конкурентоспособности на рынке труда, но и со стремительной сменой технологических процессов. Так, например, при подготовке специалистов плавсостава приходится реализовывать и национальные требования, и требования Международной морской организации (IMO) в части обязательной подготовки на современных тренажерных комплексах. Это – электронная картография, связь, средства автоматической радиолокационной прокладки и другая специализированная подготовка. Все это подразумевает высокий уровень знаний, умений и навыков в области компьютерной техники и информационных технологий, достичь которого трудно без освоения таких дисциплин, как информатика и информационные технологии профессиональной деятельности.

LXF: **Какие программные пакеты используются в процессе обучения?**

ДВМ: Перечислять полностью, наверное, нет большого смысла, так получится целый обзор, да и Linux Format неоднократно уделял значительное внимание

этой теме. В конечном итоге сам преподаватель вправе решить, какие программные пакеты использовать, условие одно – выполнение требований образовательного стандарта и рабочей программы.

Лично я строю образовательный процесс по дисциплинам «Компьютерная графика» и «Интернет-технологии» с использованием следующих программ: растровый редактор – *GIMP*; векторный редактор – *Inkscape*; программа по работе с фотографиями – *Picasa*; HTML-редакторы – *Kompozer* (бывший *Nvu*) [а точнее, его неофициальная «реинкарнация» – *прим.ред.*], *Amaya*, *Quanta Plus*; браузеры – *Mozilla Firefox*, *Konqueror*, Opera, а также ряда других. Как видите, большинство используемых программ кроссплатформенные и могут с успехом применяться не только в Linux.

Сейчас портируем из Windows программу *Ассистент 2* для проведения сетевого компьютерного тестирования; есть кое-какие нестыковки, но приобретенная программа *WINE@Etersoft 1.0 Network* должна исправить положение. Необходимость переноса программы вызвана тем, что за годы наработана огромнейшая база тестовых заданий именно под эту оболочку. В GNU/Linux есть подобные программы, например, *keduka*, *iTest*, но они пока еще уступают и по возможностям, и по удобству формирования тестового задания.



› **Преподаватель Дмитрий Валентинович Жигалов ведет занятие по информатике.**

LXF: По каким причинам возник вопрос перехода на GNU/Linux?

ДВМ: Основная причина, думаю, всем очевидна – статья 146 УК РФ. Вторая причина – коммерческая, все остальное – следствие, куда можно отнести и патриотический порыв (негоже финансировать экономику другого государства за счет своих бюджетных средств), и информационную безопасность, и тому подобное. Переход на GNU/Linux позволит раз и навсегда о них забыть, по крайней мере, о первой.

LXF: Ваше Котласское речное училище – стопроцентно государственное учебное заведение. Был ли выбор в пользу GNU/Linux как-то связан с этим фактом?

ДВМ: Котласское речное училище действительно финансируется из федерального бюджета. Конечно, как и ряд других учебных заведений, финансируемых из бюджета, мы оказываем дополнительные платные образовательные услуги в рамках действующей лицензии, но этих средств крайне недостаточно для приобретения проприетарного программного обеспечения.

Мониторинг требуемого объема лицензионного программного обеспечения, по ценам весны 2007, только по компании Microsoft составил 1,5 миллиона рублей, и это не считая других разработчиков. Общая потребность в лицензионном ПО составила более 4,0 миллионов рублей.

Когда в правительстве обсуждают затраты на легализацию ПО в школах, то почему-то забывают о профессиональных училищах, техникумах, колледжах и вузах. А ведь затраты этих учреждений, подавляющее большинство из которых – бюджетники, в несколько раз превышают затраты любой школы. Подобные затраты для нас не только неприемлемы, но и неоправданны, я бы даже сказал, расточительны.

LXF: Составлялся ли план миграции или все происходило стихийно?

ДВМ: Конечно, все было распланировано, ведь хороший план – это залог успеха. Перед тем как начать миграцию на GNU/Linux, был проработан достаточно большой объем информации: сторонний опыт, этапы, проблемы, пути реализации, дистрибутивы, пособия, методическая документация и многое другое.

Естественно, что в данный процесс были вовлечены ведущие преподаватели и заведующие лабораторий информатики, и, конечно, не обошлось без понимания и одобрения администрацией училища. Так, к примеру, директор училища, Василий Валентинович Петухов, первым высказал желание апробировать Linux на своем рабочем месте, заявив: «Справлюсь я – и другие должны справиться». Сейчас уверенно работает в Linux. Более того: предпочел установить на своем ноутбуке Linux вместо Windows.

Переход на GNU/Linux начали осуществлять по двум направлениям: первый – компьютеры администрации, второй – учебные компьютеры в лабораториях и кабинетах.

На текущий момент из 155 единиц компьютерной техники, 95 учебных и 15 административных компью-



› **Дмитрий Владимирович Моденов, зам. директора по ИТО.**

теров, полностью переведены на Linux, что составляет 70% парка.

Миграция все еще продолжается, но абсолютно полного перехода не получится, так как в образовательном процессе активно используется специализированная тренажерная техника, которая стоит несоизмеримо выше издержек на лицензии ОС Windows. К примеру, стоимость комплексного навигационного тренажера, который использует специальное ПО под Windows NT (консультировались с разработчиком, и он оказался не в состоянии предоставить версию ПО для Linux), составляет 1,8 миллиона рублей.

LXF: **Чего опасались сильнее всего?**

ДВМ: Более всего в процессе перехода опасались непонимания со стороны большинства преподавателей, которые активно используют компьютерную технику в образовательном процессе. Как ни странно, многие отнеслись к проблеме с пониманием, более того, начали осваивать новую ОС.

LXF: **Какой был выбран дистрибутив GNU/Linux и почему?**

ДВМ: Вопрос с дистрибутивом стоял наиболее остро. Среди множества достойнейших были выбраны и приобретены два дистрибутива: 25 лицензий Linux XP и Mandriva Linux 2007 Powerpack+ Academ BOX.

Выбор был сделан в пользу этих дистрибутивов, потому что Linux XP имеет переработанный рабочий стол Gnome, напоминающий знакомый многим интерфейс (сила привычки, и от этого никуда не деться), невысокие аппаратные требования, совместимость с имеющимся оборудованием, улучшенную поддержку кириллицы в приложениях, мультимедийность и, конечно, неплохой сайт поддержки от разработчика. Данный дистрибутив устанавливается только на компьютеры администрации.

Mandriva Linux 2007 Powerpack+ Academ BOX – лицензия на неограниченное количество ПК в учебном заведении, возможность установки на ПК преподавателей и студентов, класс инсталляции рабочего стола и сервера, дружественный интерфейс (используем рабочий стол KDE), расширенная поддержка оборудования, распространенность ОС в России, большой репозиторий, отличная панель управления, мультимедийность, наличие учебных пособий. Данный дистрибутив устанавливается на учебных компьютерах.

В дополнение к дистрибутиву Mandriva 2007 Powerpack+ приобретена сетевая версия *Wine@Etersoft 1.0 Network*. Стоимость решения составила 49,1 тысячи рублей.

LXF: **Справляетесь?**

ДВМ: Лукавить не буду – стало сложнее, но это сейчас. Собственных сил хватает, а вот знаний и опыта еще не достаточно. Опыт приходит только со временем. В связи с этим хотелось бы иметь оперативную и качественную, пусть даже платную поддержку.

LXF: **Как на GNU/Linux отреагировали учащиеся?**

ДВМ: По моим наблюдениям, учащимся нет большой разницы, что изучать. Для многих, особенно первокурсников, тут кнопки – там кнопки...

Иначе обстоят дела со студентами специальности «Автоматизированные системы обработки информации и управления», ведь это их будущий «хлеб». Специалист, владеющий навыками администрирования Linux, примерно в 1,5 раза более высокооплачиваем в сравнении с аналогичным, но способным администрировать только Windows. Многие проявляют интерес к различным дистрибутивам GNU/Linux, и не только. Некоторые даже заинтересовались ОС Solaris и FreeBSD.

Ряд студентов использует на домашних компьютерах GNU/Linux, но большинство на сегодня используют ОС Windows, и это прежде всего связано с отсутствием качественных игр под GNU/Linux. Знаю, что ситуация и в этом направлении начинает меняться, выходят специальные игровые дистрибутивы, но для всего нужно время.

LXF: **Были ли при переходе проблемы социального плана у учителей и администрации? Как их удалось решить?**

ДВМ: Личный пример, убеждение и плечо товарища всегда помогут в подобных ситуациях. Я уже приводил пример, как наш директор филиала начал с себя – другие комментарии, думаю, не нужны.

Проводилась разъяснительная работа в коллективе на темы статьи 146 УК РФ, о преимуществах программ с открытым кодом, а также консультационная помощь. Был приобретен и опубликован в интранет ряд курсов Интернет-Университета Информационных Технологий (http://www.intuit.ru): «Основы работы в ОС Linux», «Операционная система Linux», «Основы работы в OpenOffice».

LXF: **Какие методические материалы сейчас используются?**

ДВМ: Основные методические материалы, которые используются нами сейчас – ресурсы сети интернет:

- http://intuit.ru – Интернет-университет информационных технологий
- http://www-128.ibm.com/developerworks/ru/linux/ – сайт компании IBM
- http://www.linuxcenter.ru/lib/books/ – сайт компании LinuxCenter
- http://linux-admin.net.ru/

Здесь перечислены далеко не все используемые нами источники информации в сети. Ну и нельзя не упомянуть в этой связи журнал Linux Format.

LXF: **Что бы вы пожелали сообществу для того, чтобы процесс внедрения GNU/Linux в образование пошел активнее?**

ДВМ: Для начала необходимо ликвидировать пробел с качественными учебниками и учебными пособиями, и не только по конкретному дистрибутиву GNU/Linux, но и по работе с программными продуктами. Сейчас зачастую приходится собирать нужную информацию из разных мест. Сама информация об одном и том же предмете может сильно различаться, а иногда и вовсе не соответствовать действительности, или попросту быть устаревшей. Необходимо больше уделять внимания качеству локализации продуктов. Было бы здорово, если бы разработчики улучшили поддержку пользователей с сайта по различным вопросам инсталляции и настройки системы, настройки конкретного оборудования, установки и удаления программных пакетов, и т.д.

LXF: **Вы упомянули журнал Linux Format. Чем он вам нравится? Что бы вы ему пожелали?**

ДВМ: Прежде всего, приятно чувствовать, что ты не один, думаю, вы понимаете, о чем я [*улыбается*]. Обзоры программных пакетов, публикуемые в Linux Format, позволяют сильно экономить время, а советы от профессионалов и учебные пособия просто бесценны! Свежие релизы на прилагаемом DVD позволяют не только держать руку на пульсе, но и экономить средства. Это особенно актуально, если не имеешь качественного канала связи. Пожелание только одно – так держать! LXF



› **С.П. Кубраков проводит занятие по тестированию аппаратной части.**



GNU/Linux покоряет Уфу

По традиции, технари скептически относятся к гуманитариям. Но раз уж гуманитарии существуют, и их много, остается их просвещать. Этим и занимается **Азат Салаватович Канчурин** из города-миллионика Уфа.

Что бы делал без него в освоении Линукса — неизвестно!
Отзыв на http://ufa.prepod.org

Евгений М. Балдин, LXF: **Представьтесь, пожалуйста, Азат Салаватович.**

Азат С. Канчурин: Я являюсь доцентом кафедры информационных технологий Восточного института экономики, гуманитарных наук, управления и права (ВЭГУ), г. Уфа. В данном вузе преподаю уже пятый год – можно сказать, «старожил» [*улыбается*]. Сам уже лет семь как закончил учебу в Башкирском государственном университете. После учебы некоторое время трудился на одном из малых предприятий нашего большого города [*улыбается*], где понял, что мне проще работать с заинтересованными людьми, нежели с коллегами, у которых в жизни только один интерес – «от зарплаты до получки». Более того, склад моего характера не позволяет мне постоянно заниматься одним и тем же... Это утомляет... В свою очередь, работа в вузе, помимо того, что постоянно заставляет быть в форме, еще и не позволяет скучать – каждый день что-то новое: занятия наукой, решение прикладных проблем, воспитательная работа (представьте себе, такое тоже есть), и это все помимо преподавания.

LXF: **Кого готовит ваша кафедра?**

АСК: Сегодня работодателю нужно, чтоб наш студент «на выходе» знал, чем «напичкан» компьютер и как при случае чинить его собственными силами, умел программировать на высокоуровневом языке (C/C++ либо среда Delphi), прокладывать и настраивать сети (любой морфологии), создавать базы данных и работать с ними. Не буду пересказывать государственные требования, замечу лишь, что кафедра стремится лавировать меж двух огней: и государство «не обидеть», так как реалии нашей жизни часто идут вразрез с образовательным стандартами, и в тоже время подготовить к самостоятельной профессиональной жизни высококлассных специалистов.

Помимо того, что мы ведем дисциплину «Информатика» на всех факультетах вуза (в академии учатся юристы, экономисты, педагоги, психологи, социальные работники, филологи, журналисты, так что работы хватает), преподаватели кафедры руководят дипломными проектами студентов специальности «Информатик-экономист».

LXF: **По каким причинам возник вопрос перехода на GNU/Linux?**

АСК: Думаю, не открою Америку, ответив, что до последнего времени на кафедре использовался Windows. Лично я пользовался в образовательном процессе такими пакетами, как *MatLab*, *MathCAD*, *Visual Studio*, виртуальными машинами. Другие педагоги используют 1С, продукты IBM, кое-что от Microsoft.

Так уж случилось, что организация работы в



Windows мне изначально не нравилась. Для расширения кругозора (да и из здорового интереса) попеременно устанавливал себе на домашнюю машину различные версии GNU/Linux, пока не появилась Mandriva 2007.0. «Влюбился» в нее, можно сказать, с первых кликов мыши. Мало того, что Free-версия устанавливала все ПО «первой необходимости» прямо «из коробки» без лишних телодвижений, так еще у меня все прекрасно работало...

Как ни странно это звучит, но полностью перевести свой учебный процесс на GNU/Linux меня заставили игры, то есть не сами игры как «субъект», а острая и постоянная тяга к ним наших студентов. Раньше после раздачи самостоятельных заданий для выполнения за рабочим местом мне приходилось стоять, что называется, над душой каждого учащегося, чтоб он хоть как-то что-то начинал делать. Стоило мне отвернуться, отвлечься, выйти из класса, как товарищи начинали «резаться в Контру». Чего я только ни делал: и удалял в начале рабочего дня со всех машин хоть какой-то намек на игрушки, и выдирал волокно из хабов, и ставил пароль на установку программ – ничего не помогало, через пятнадцать минут любая «защита» ломалась [*улыбается*].

После некоторых колебаний, которые длились два года, я решился на радикальное вмешательство. Снес весь Windows со всех машин и установил Mandriva Linux 2007.0 Free. Народ, естественно, «взвыл», однако студенты потихоньку начали «забывать», где находятся компьютерные классы с играми, а преподаватели постепенно начинают привыкать к новому формату работы.

LXF: Какие программы используются при обучении? Как протекает сам процесс?

АСК: Сегодня первокурсники тратят в среднем полчаса в начале семестра на усвоение команд создания каталога и текстового файла, перехода из каталога в каталог и компиляции программ. Для предметов, на которых изучаются высокоуровневые методы программирования и проектирования ПО, мы используем библиотеку *Qt* и среды *KDevelop*, *Eclipse*, для работы в сетях и настройки сервера *Apache* мой коллега применяет язык сценариев Perl в связке с PHP. Есть у меня курс интеллектуальных технологий, под который адаптирую пакет *SciLab* как альтернативу *MatLab*; возможно, в дальнейшем буду знакомить студентов с интерпретатором *swi-Prolog*. Несомненным подспорьем для нас станет вышедший недавно первый релиз EduMandriva, предназначенный специально для образовательных учреждений, созданный Александром Казанцевым (LXF97) и Центром компетенции Mandriva.ru города Глазова, с которым наша кафедра поддерживает тесные связи. Естественно, под каждую дисциплину сегодня готовятся отдельные базы учебно-методической литературы, которые сразу же выпускаются издательством нашего вуза.

LXF: Как ваши действия вписываются в образовательную политику вашего вуза?

АСК: Стоит отметить всяческую поддержку со стороны руководства в направлении перспективного развития Open Source. Сегодня на базе нашего факультета открывается лаборатория Открытого программного обеспечения и сетевых технологий, в которой ведется работа по планомерному переходу всего вуза на открытое ПО, созданию «стендов» производственных предприятий, с целью дальнейшего выхода на потенциальных заказчиков; в рамках сетевых технологий планируем развитие Wi-Fi- и Wi-Max-коммуникаций на платформе GNU/Linux – для этого уже заключены договоренности с представительством фирмы D-Link в Уфе на предоставление оборудования и технологий.

Несмотря на то, что вуз наш считается полностью коммерческим предприятием, он является своего рода флагманом в инновационных сферах образования, экспериментальным «полигоном», результатами работы которого становятся востребованные кадры, выпускаемые вузом. Сегодня витают идеи использовать GNU/Linux в сфере дистанционного образования. Это накладывает на кафедру и лабораторию дополнительные обязательства подготовки экстерн-курсов безболезненного перехода на GNU/Linux «на расстоянии». Кроме того, вуз обязательно запустит в ближайшем будущем курсы для преподавателей школ и всех желающих по миграции на GNU/Linux. Планов громадье...

LXF: Был ли план миграции или все делалось стихийно?

АСК: План миграции был с самого начала; правда, в первой редакции предполагалось перевести на GNU/





Linux весь вуз разом. Подумав, мы все же решили пока «не пить шампанское» и подошли к решению проблемы с другой стороны: полный переход на GNU/Linux одного базового факультета, с дальнейшим анализом наработанного опыта (да и лаборатория «в подмогу») и постепенным переводом всей компьютерной техники на открытое ПО.

LXF: Какой был выбран дистрибутив GNU/Linux и почему?

ACK: Дистрибутив – Mandriva Linux 2007.0 Free. Сейчас подумываю о 2007.1, тем более что EduMandriva был собран под нее.

LXF: Как бы вы отнеслись к платной поддержке?

ACK: Платная поддержка? Я просто не представляю ее себе. Ведь в Уфе (к огромному сожалению) нет представительств Mandriva (да и других дистрибутивов тоже). Не думаю, что приезд специалиста из Франции (или Санкт-Петербурга) будет решением всех проблем, а вступать в оплачиваемую переписку не вижу смысла, при наличии Google, книг и головы [*улыбается*].

Но в любом случае хотелось бы, чтоб представительство Mandriva в России задумалось об открытии Центра компетенции в Уфе – тем более, «задел» в виде наработок нашей кафедры и спрос среди предприятий республики имеется.

LXF: Как на GNU/Linux отреагировали студенты?

ACK: Один-два раза я мельком слышал от студентов ропот: «Почему не винь?». Но слабые попытки «проявить» полностью характер потребителя потерпели фиаско под «лавиной» «авторитета толпы», а может, и моего [*улыбается*]. Наш студент всегда был готов к различного рода новшествам, и привык принимать их с удовольствием: ведь что может быть лучше познания непознанного?

Сегодня меня некоторые еще называют «монстром» и «извергом», но постепенно это сходит на нет, потому что с GNU/Linux'ом учащимся реально интереснее работать: нетривиальные задачи, которые система подчас ставит перед нами, заменяют нам «полевые условия» любого предприятия. Могу с уверенностью сказать, что сегодня студенты прониклись уважением к этой дружественной системе, чувствуют себя особой «кастой». Пусть мы еще не много знаем о ней, пусть у нас не все получается, но от этого наша жизнь только интереснее и насыщенней. Большая часть студентов уже установила GNU/Linux на своих домашних компьютерах, потому как мы иногда задаем работу на дом.

LXF: Как из-за перехода на свободное программное обеспечение изменился образовательный процесс?

ACK: Сказать, что ничего не изменилось – значит слукавить. Но с высоты прошедшего времени кажется именно так. А вообще, изменился сам подход к преподаванию со стороны преподавателей и к усвоению материала студентами. Проще не стало, это было бы скучно... [*улыбается*]

LXF: Чего вам сейчас не хватает для полного счастья?

ACK: Как всегда, времени: 24 часа в сутки – это очень мало.

Для непосредственной работы с GNU/Linux еще бы пару свободных классов для экспериментов, потому как имеющиеся с утра до вечера бывают полностью заняты выведением их из строя [*улыбается*].

LXF: Что бы вы пожелали сообществу, чтобы процесс внедрения GNU/Linux в образование пошел активнее?

ACK: Не бойтесь ничего, экспериментируйте, не забывайте, что компьютер – это всего лишь десяток электронных «штучек», соединенных между собой в определенном порядке, и нужно очень сильно постараться, чтоб вывести его из строя. А вот чтобы вывести из обоймы студентов, достаточно пары «несерьезных» скучных занятий. К счастью, сегодняшний GNU/Linux при умелом его использовании способен решать и эти проблемы.

LXF: Вы можете что-нибудь сказать про Linux Format?

ACK: Конечно! Удивительно, но наткнулся я на него при совершенно невообразимом стечении обстоятельств. Это тема отдельного разговора.

Журнал произвел на меня огромное впечатление: доброжелательностью и толерантностью, обилием готовых решений, примеров, «заквасок для ума». Отличное начинание, отдельное спасибо его издателям и всем тем, кто делает нашу жизнь ближе к Linux. LXF



Журнал зарегистрирован Федеральной службой по надзору за соблюдением законодательства в сфере массовых коммуникаций и охране культурного наследия
ПИ № ФС77-21973 от 14 сентября 2005 года
Выходит ежемесячно. Тираж 6000 экз.

РЕДАКЦИЯ РУССКОЯЗЫЧНОЙ ВЕРСИИ:
ГЛАВНЫЙ РЕДАКТОР
Валентин Синицын info@linuxformat.ru
Литературные редакторы
Родион Водейко, Елена Толстякова, Иван Мищенко
Переводчики
Илья Аввакумов, Александр Бикмеев, Светлана Кривошеина, Александр Казанцев, Алексей Опарин, Валентин Развозжаев, Татьяна Цыганова, Юлия Шабунио
Редактор диска
Александр Кузьменков
Допечатная подготовка
Мария Пучкова, Родион Водейко
Креативный директор
Станислав Медведев
Технический директор
Денис Филиппов
Директор по рекламе
Денис Игнатов +7 812 965 7236 advert@linuxformat.ru
Заместитель генерального директора
Софья Виниченко
Генеральный директор
Павел Фролов
УЧРЕДИТЕЛИ
частные лица
ИЗДАТЕЛИ
Станислав Медведев, Виктор Федосеев, Павел Фролов

Отпечатано в типографии «Текст», ООО «ППК «Текст»
188680, Ленинградская область, Всеволожский район, Колтуши, д.32
Заказ _____
Пре–пресс: *d.r.i.v.e-group*

РЕДАКЦИЯ АНГЛОЯЗЫЧНОЙ ВЕРСИИ:
Редактор Ник Вейч [Nick Veitch] nick.veitch@futurenet.co.uk
Заместитель редактора Пол Хадсон [Paul Hudson] paul.hudson@futurenet.co.uk
Художественный редактор Эфрейн Эрнандес-Мендоса [Efrain Hernandez-Mendoza] efrain.hernandez-mendoza@futurenet.co.uk
Редактор новостей Майк Сондерс [Mike Saunders] mike.saunders@futurenet.co.uk
Редактор обзоров
Грэм Моррисон [Graham Morrison] graham.morrison@futurenet.co.uk
Литературный редактор
Мэтт Нейлон [Matt Nailon] mnailon@futurenet.co.uk

Подготовка материалов
Ладислав Боднар [Ladislav Bodnar], Нейл Ботвик [Neil Bothwick], д-р Крис Браун [Dr. Chris Brown], Энди Ченнел [Andy Channelle], Крис Хауэллз [Chris Howells], Евгений Балдин, Александр Бабаев, Андрей Боровский, Андрей Паскаль, Галина Пожарина, Александр Поносов, Валентин Развозжаев, Александр Супрунов, Алексей Федорчук, Илья Шпаньков

Художественные ассистенты: Ник Кокс [Nick Cox], Ники Гоутубед [Nicky Gotobed], Дэмиан МакГи [Damian McGee]
Фотографии: Джена Камбо [Jena Cumbo], iStock Photo
Иллюстрации: Пол Блечфорд [Paul Blachford], Крис Винн [Chris Winn]
Обложка и иллюстрации к спецрепортажу: Радим Малинич [Radim Malinic]

КОНТАКТНАЯ ИНФОРМАЦИЯ
UK: Linux Format, 30 Monmouth Street, Bath BA1 2BW
Tel 01225 442244 Email: linuxformat@futurenet.co.uk
РОССИЯ:
Санкт-Петербург (редакция): ул. Гончарная, 23, офис 54, телефон: (812) 717-00-37
Email: info@linuxformat.ru, Web: www.linuxformat.ru
Представительство в Москве: ул. Энергетическая, д.14, корпус 5, стр. 1
По вопросам сотрудничества, партнерства, оптовых закупок: +7 (495) 799-18-63
Дирекция московского офиса: +7 (495) 136-88-45. E-mail: moscow@linuxcenter.ru





Дорога в будущее

Мы поговорим с людьми, разрабатывающими ваше любимое ПО, а они расскажут, почему перспективы для Linux никогда еще не были столь радужны.

▸▸ Хакерство с Arduino

Узнайте, как собрать свое собственное устройство и запрограммировать его – независимо от вашего опыта кодирования.

▸▸ Суперобзор дистрибутивов

Ubuntu против Mandriva против OpenSUSE: что больше подходит именно вам?

▸▸ Сравнение: Подкасты для Linux

Музыка для ушей любителя свободного ПО.

Содержание последующих выпусков может меняться без уведомления